UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

SISTEMAS DE INFORMAÇÃO

MAURICIO BRANCO BIAZUS

Subvertendo um sistema de detecção de intrusão: Caso prático utilizando Snort e Nmap

TRABALHO DE CONCLUSÃO DE CURSO

Florianópolis

2016

MAURICIO BRANCO BIAZUS

Subvertendo um sistema de detecção de intrusão: Caso prático utilizando Snort e Nmap

Trabalho de conclusão de curso apresentado ao curso de Sistemas de informação, como parte dos requisitos necessários à obtenção do título de Bacharel em Sistemas de informação.

Orientador: João Bosco Mangueira Sobral

Florianópolis
2016

Mauricio Branco Biazus

Subvertendo um sistema de detecção de intrusão: Caso prático utilizando Snort e Nmap

Este trabalho de Conclusão de Curso foi julgado, adequado e aprovado em sua forma final para obtenção do grau de Bacharel em Sistemas de Informação da Universidade Federal de Santa Catarina.

Florianópolis, ____ de _____ de _____.

Orientador :

_____

Prof. Dr. João Bosco Mangueira Sobral

Banca avaliadora:

_____

Profª. Dr. Carla Merkle Westphall

_____

Prof. MSc. Fernando Augusto da Silva Cruz

Florianópolis

2016

À minha mãe;

Minha razão e inspiração.

Agradecimentos

Agradeço primeiramente aos meus pais, Meri de Oliveira Branco e Angelo Ary Biazus, e minha irmã, Bruna Branco Biazus, pelo amor e apoio incodicionais.

Agradeço à todos os professores e colaboradores da UFSC e do curso de Sistemas de Informação, pelos conhecimentos transmitidos e pelo suporte para meu crescimento acadêmico.

Agradeço ao professor Dr. João Bosco Mangueira Sobral, pela orientação e suporte durante todo o desenvolvimento deste trabalho.

Agradeço aos meus irmãos que o vôlei me proporcionou, Luiz Otavio Polanski Paese, Jefferson Isberner Santana, Wagner Samuel Fabrício e até ao Moisés Felipe Teixeira.

Agradeço aos meus irmãos que Palmas me deu, Ruan Aquino, Gabriel Mendes Folha Seca e Hector Jacques.

Agradeço à minha família de Concórdia, que de longe, me dá inspiração para a vida em família: Vó Audi, Vô Octa, todos os Biesus e agregados, especialmente para o Tio Paulão, Tia cleo, William e Filipe.

Agradeço à Carolina Lopes Monteiro pelo apoio, companheirismo e paciência.

Agradeço à todos, que direta ou indiretamente, contribuíram para meu desenvolvimento neste período.

Change will not come if we wait for some other person or some other time. We are the ones we‘ve been waiting for. We are the change that we seek.
Barack Obama

Resumo

O Sistema de detecção de Intrusão (IDS) é uma camada de segurança que, a partir da análise e monitoramento de eventos, na rede ou em seu hospedeiro, gera alertas a partir de regras pré-estabelecidas afim de avisar os responsáveis e facilitar a tomada de decisão e contramedidas. Um dos mais utilizados IDS pela comunidade é o Snort, que tem seu código aberto e é baseado em regras muitas delas estabelecidas pela própria comunidade.

No contra fluxo, estão os exploradores de vulnerabilidades e mapeadores de redes. Normalmente se baseiam no envio de pacotes afim de conseguir informações relevantes para ataques. O mais conhecido mapeador de rede é o Nmap. O Nmap, como o Snort, é de código aberto mantido por uma relativamente grande comunidade.

Mesmo as duas ferramentas sendo amplamente conhecidas, seus limiares trazem, para quem as usa, brechas de acordo com as configurações estabelecidas.

Este trabalho de conclusão de curso tem como objetivo explorar alguns dos limiares padrão do Snort, utilizando para isso o Nmap, sempre com a premissa de tentar esconder a identidade do atacante ou distraí-lo. Para que baseado nestes ataques, possa ser feita uma auditoria de segurança na rede monitorada.

Palavras-chave: Snort, Nmap, Sistema de detecção de intrusão, IDS, Intrusion Detection System, Subversão de IDS

Abstract

The Intrusion Detection System (IDS) is a security layer that, from the analysis and monitoring of events, on the network or its host, generates alerts from pre-established rules in order to warn those responsible and facilitate the taking And countermeasures. One of the most used IDS by the community is Snort, which has its code open and is based on rules many of them set by the community itself.

In the counterflow, are vulnerability exploiters and network mappers. They are usually based on sending packets in order to get information relevant to attacks. The best-known network mapper is Nmap. Nmap, like Snort, is open source maintained by a relatively large community.

Even though the two tools are widely known, their thresholds bring, for those who use them, loopholes according to the established settings.

This course completion work aims to explore some of the standard Snort thresholds, using Nmap, always with the premise of trying to hide the identity of the attacker or distract him. So that based on these attacks, a security audit can be done in the monitored network.

Keywords: Snort, Nmap, Intrusion Detection System, IDS, Intrusion Detection System, IDS Subversion

Lista de ilustrações

Lista de abreviaturas e siglas

CTC	Centro Tecnológico
CUI	Console User Interface
FTP	File Transfer Protocol
GUI	Graphic User Interface
HIDS	Host Intrusion Detection Service
ICMP	Internet Control Message Protocol
ID	Identificador
IDS	Intrusion Detection System
IP	Internet Protocol
LOG	Registro de evento
NIDS	Network Intrusion Detection Service
SNMP	Simple Network Management Protocol
SO	Sistema Operacional
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UFSC	Universidade Federal de Santa Catarina

Sumário

1 Introdução

A segurança na Internet sempre foi motivo de preocupação para todos os usuários, tanto para usuários domésticos, quanto para usuários corporativos e com o aumento exponencial das redes e do seu uso, as atenções estão cada vez mais voltadas para os campos de prevenção, contenção e contra-ataque as ameaças e usuários maliciosos. A lista de ameaças inclui vírus de computador, worms, spywares, exploits e os próprios Crackers. Esta lista só não é maior do que a de ferramentas e técnicas utilizadas para tentar conter essa avalanche de ameaças: firewalls, antivirus, anti-spywares, anti-rootkits, verificadores de integridade, configuração segura de servidores, entre muitos outros. Nesta lista de armas para manter o nível de segurança aceitável estão os Sistemas de Detecção de Intrusão, que têm como princípio básico procurar antecipar possíveis tentativas de acesso malicioso através de monitoramento e detecção de padrões na rede de ataques conhecidos ou de mudanças no padrão de comportamento, do uso da rede.

> Segundo Barford et al. (2002 apud AZEVEDO, 2012, p. 17), “as redes de computadores sem análise de tráfego não podem operar eficientemente ou com segurança.”

A análise do tráfego de uma rede é fundamental para manutenção de um bom funcionamento da rede, tanto em questão de segurança como também em questão de qualidade, já que ataques também podem comprometer direta ou indiretamente o desempenho do ambiente onde está ocorrendo.

O Snort passa a ser objeto de estudo deste trabalho de conclusão de curso pois além de ser um dos mais conhecidos Sistemas de detecção de intrusão, via de regra possui os mesmos pontos fortes e fracos de outros sistemas. Este IDS é bastante usado e por isso mesmo tem uma grande comunidade que dá suporte ao software, mesmo por que é de distribuição *open source*, e junto com essa grande quantidade de pessoas que dão suporte, vem junto uma grande quantidade de pessoas que tentam subverter o Snort. E por ser *open source*, e se basear em regras conhecidas, abre brecha para que sejam exploradas falhas nas configurações que normalmente não são reguladas ou personalizadas, e então são deixadas no modo padrão pelos usuários finais.

Tendo em vista estas vulnerabilidades, elaboram-se os seguintes objetivos, os quais nortearam o desenvolvimento do trabalho:

1.1 Objetivo geral:

Subverter uma instância do Snort baseando os ataques nas instruções coletadas do livro “Exame de redes com Nmap” (LYON, 2009) e no referido software de mapeamento de rede, Nmap, a partir de uma máquina dentro da rede interna monitorada pelo IDS.

1.2 Objetivos específicos:

Os objetivos específicos que podem ser citados são:

- Compreender os requisitos, características, modos de atuação e funcionalidades do IDS Snort e do mapeador de redes Nmap
- Mapear os principais ataques cometidos atualmente nas tentativas de mapeamento de portas.
- Monitorar o comportamento do Snort e suas regras na detecção destes ataques partidos do Nmap.
- Apresentar possíveis estratégias de ataque para evitar alertas pelo Snort, para que, baseados neles, se possa melhorar a estratégia de filtragem do Snort.

1.3 Motivação

Segundo (LYON, 2009):

> Um dos princípios centrais de segurança de redes é que a redução do número e da complexidade dos serviços oferecidos reduzirá a oportunidade dos atacantes irromperem.

A maioria dos comprometimentos de redes são obtidos através da exploração de uma aplicação servidora que atende a uma porta, seja ela TCP ou UDP. Muitas das vezes estas aplicações nem mesmo são utilizadas pelos reais usuários da rede, na verdade são habilitadas por omissão desde a sua configuração.

Entendendo que toda a porta aberta é uma oportunidade de comprometimento, os usuários maliciosos podem examinar a rede, à procura de alvos, e neles fazer varreduras regulares para listar as portas abertas. tendo esta informação em mãos, é possível comparar esta lista com os serviços vulneráveis e explorar suas brechas e comprometer a máquina alvo. Uma importante defesa contra este tipo de ataque é os administradores de sistemas examinarem suas próprias redes regularmente, com ferramentas como o Nmap. A partir dele, o responsável pode obter a lista das portas abertas e desativar quaisquer serviços que não sejam utilizados.

Mas, no caso de os serviços estarem sendo utilizados, eles não podem simplesmente ser desabilitados. Nestes é necessário fazer o monitoramento através de Sistemas de detecção de intrusão como o Snort. Porém da mesma forma que as máquinas sofrem por omissão de configuração, o Snort (ou qualquer IDS) sofre do mesmo mal. É para isso que deve ser usado o Nmap para testar as regras do Snort, e assim ajustá-las e criar além do que as disponíveis por padrão, e assim, será possível perceber ataques aos serviços ativos.

1.4 Trabalhos relacionados

Em (NUNES, 2014), o autor propõe um estudo para testes de penetração em uma rede, onde teste de penetração é definido como:

> O teste de penetração é um método que avalia a segurança de um sistema de computador ou de uma rede, simulando um ataque do mundo real. O processo envolve uma análise nas atividades do sistema, que envolvem a busca de alguma vulnerabilidade em potencial que possa ser resultado de uma má configuração do sistema, falhas em hardwares/softwares desconhecidas, deficiência no sistema operacional ou técnicas contramedidas.

(SILVA, 2015) a partir do mesmo princípio proposto por (NUNES, 2014) propõe testes de penetração, mas focados em ultrapassar firewalls

O objetivo era simular ataques com o intuito de mensurar o impacto desses, caso sejam bem-sucedidos. Desta forma seria possível descobrir vulnerabilidades, identificar os riscos que podem ser difíceis de detectar, testar a capacidade defensiva da rede e identificar a reação do sistema aos ataques. Dentre vários motivos para realizar um ataque, destacam-se as invasões por questões financeiras, pessoais, fraudes, sabotagem ou espionagem. O invasor é uma pessoa com alto nível de conhecimento técnico, seus ataques são minuciosamente planejados, e é importante que haja o estudo do comportamento do alvo, assim podendo-se descobrir brechas na segurança.

Em ambos os trabalhos, o invasor é considerado uma ameaça externa, que tem que passar pela Internet, seus firewalls e roteadores com filtragem de pacotes, para enfim, chegar ao alvo. Este presente trabalho de conclusão de curso porém, considera o invasor como uma ameaça interna, dentro da rede monitorada pelo Snort, desconsiderando assim qualquer influência externa da Internet, como firewall externo ou roteadores com filtragem de pacotes. Este tipo de situação ocorre em locais como na Universidade Federal de Santa Catarina, em qualquer um de seus laboratórios de pesquisa. E assim, este trabalho almeja complementar os estudos citados anteriormente na área de segurança e monitoramento contra invasões de redes.

1.5 Organização do Trabalho

No texto que segue, o capítulo 2 trata da fundamentação teórica para o que são e como funcionam os sistemas de detecção de intrusão em geral. O capítulo 3 mostra especificamente as características do IDS Snort, como a sua forma de classificação de alertas. Em seguida, no capítulo 4, é feito o detalhamento da instalação e configuração do Snort feita para a execução deste trabalho. Já no capítulo 5 trata do explorador de redes Nmap e sua versão gráfica, o Zenmap. Cita os principais ataques, suas possíveis motivações e detalhes. Na sequência está o capítulo 6 onde são exibidas teorias de como detectar sistemas de detecção de intrusão em uma rede. No capítulo 7 são apresentados os dados relacionados aos testes dos próximos capítulos. O capítulo 8 tem o início das experiências práticas, neste caso, sendo elas com o propósito de evitar a detecção por parte do IDS. Em seguida, no capítulo 9 são apresentadas outros experiências práticas, porém com o propósito de distrair um IDS. Finalmente no capítulo 10 está a conclusão do trabalho, seguido então pelos trabalhos futuros.

2 Sistemas de Detecção de Intrusão

A utilização de métodos de detecção de intrusão permite recolher informação sobre tipos de ataques já conhecidos e identificar tentativas de ataque à rede ou a algum servidor em particular. A informação recolhida servirá, essencialmente, para a tomada de decisões que levem à proteção do alvo do ataque e também poderá constituir uma base informativa para uma ação legal.

Um sistema global de segurança, consiste num conjunto de ferramentas que incluem:

- Firewalls
- Sistemas de Detecção de Intrusão (IDS)
- Sistemas de Avaliação de Vulnerabilidade

Estas ferramentas deverão trabalhar em conjunto e partilhar informação de uma forma dinâmica.

Os Sistemas de Detecção de intrusão, em Redes de computadores (NIDS, Network Intrusion Detection System), são utilizados para monitoramento do tráfego de dados de uma rede ou de um segmento de rede. A análise é realizada em dados coletados da rede, ou em base de dados disponíveis ao NIDS.

Os IDS tem como principais vantagens detecção e resposta em tempo real, dificuldade de remover evidências pelos atacantes por ficar em uma maquina dedicada e protegida e baixo custo. Porém, os sistemas de detecção de intrusão também contam com desvantagens, que incluem dificuldade de identificar ataques fragmentados, não ser possível analisar protocolos criptografados e possui ainda pontos cegos.

2.1 Categorias de IDS

A detecção de intrusão é um conjunto de técnicas e métodos que são utilizados para detectar atividades suspeitas, tanto ao nível de rede como no nível do servidor. Os sistemas IDS são normalmente agrupados em três categorias:

- IDS Baseado em rede
- IDS baseado em host
- IDS distribuido

A maior parte dos IDSs são passivos, com capacidade de integrarem módulos que permitam alguma reação. Os IDSs reativos são capazes de, por exemplo, “matar” uma sessão ou reprogramar a firewall de forma dinâmica.

2.1.1 IDS Baseado em rede

Sistemas NIDS monitoram toda uma rede atrás de anomalias e ataques que estão em curso. Para que ele consiga analisar os pacotes, sensores são colocados em pontos estratégicos da rede. Estes sensores, monitoram todo o tráfego de entrada e saída, atrás de payloads que possam indicar que um ataque está sendo realizado. Quando um ataque é detectado, o software avisa ao administrador e este pode tomar alguma ação logo no início do ataque. O Snort, Bro NIDS e o Suricata são exemplos de NIDS gratuitos. O Guardian é um módulo que integra com o Snort e que atualiza de forma automática as regras da firewall baseadas nos alertas gerados pelo Snort.

2.1.2 IDS baseado em host

Enquanto sistemas NIDS monitoram redes inteiras, sistemas HIDS monitoram apenas um único host na rede. Ao invés de monitorar apenas pacotes de rede, o software irá também monitorar que processo acessa qual recurso, quais arquivos são alterados, verifica as informações da RAM e logs e garante que as informações destes não foram alterados. O OSSEC e AIDE são exemplos de HIDS gratuitos.

2.1.3 IDS distribuido

Num DIDS existem vários NIDS/HIDS (sensores) com regras específicas relativamente à sua localização e os alertas são enviados para uma estação central de monitoração. A comunicação dos sensores com a estação central de monitoração deverá ser estabelecida numa rede privada ou no caso da necessidade de utilizar o mesmo segmento de rede, configurar uma VPN.

2.2 Métodos de detecção dos IDS

O método de detecção dos Sistemas de Detecção de Intrusão dividem-se em duas categorias básicas:

- Detecção baseada em assinaturas de intrusão
- Detecção de anomalia (estatística)

Uma assinatura refere-se geralmente a um conjunto de condições que caracterizam a manifestação direta de atividades de intrusão em termos de cabeçalhos de

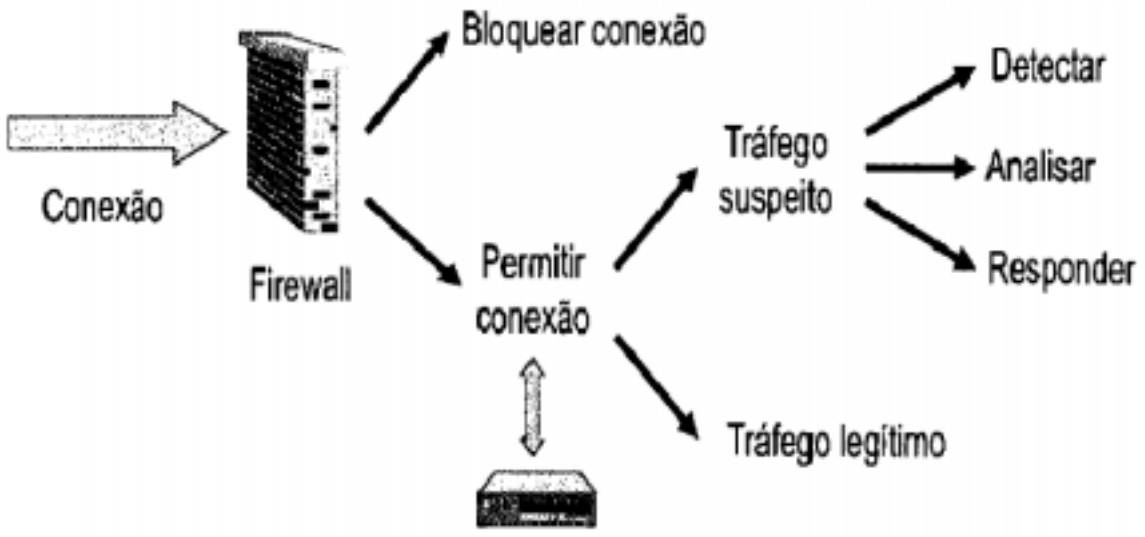
pacotes e conteúdo útil (payload). Historicamente, o método baseado em assinaturas foi o mais utilizado em termos de NIDS. Este método baseia-se na sua base de dados de assinaturas de ataques e quando uma ou mais dessas assinaturas é detectada em trânsito, é acionado um alarme e o evento registado para investigação. A robustez da detecção de intrusão baseada em assinatura, está diretamente relacionada com a qualidade e atualização da base de dados de assinaturas.

A detecção baseada em anomalias dispara um alarme quando observa um comportamento fora do comum na rede. Por si só, este tipo de método não detecta todos os tipos de ataque, mas torna-se muito eficaz quando utilizado em conjunto com métodos de detecção por assinatura. Os modelos estatísticos mais utilizados em detectores de intrusão por anomalia foram propostos em (DENNING, 1986).

2.2.1 Componentes do IDS

De acordo com Nakamura (2007), um sistema de detecção de intrusão funciona de acordo com uma série de funções que, trabalhando de modo integrado, são capazes de detectar, analisar e responder as atividades suspeitas. A figura 1 apresenta as funções e distribuição de um NIDS.

Figura 1 – Funções de um NIDS em uma rede



Funções dos NIDS. (NAKAMURA, 2007).

Como se verifica na figura 1, após o firewall liberar as conexões, os Sistemas de Detecção coletam, analisam, armazenam essas informações e respondem às atividades suspeitas, classificando o tráfego como suspeito ou legítimo. Para que o sistema tenha um bom desempenho na detecção de intrusão necessita-se um conjunto de ferramentas

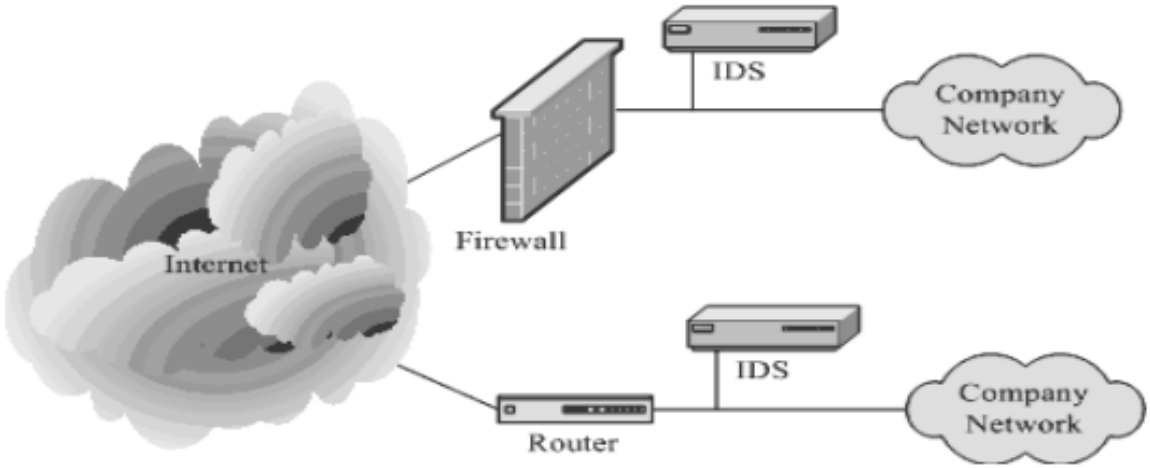
básicas como um coletor (sonda), analisador, Banco de dados (ex: Mysql), atuador e monitor, conforme especificado abaixo:

- Coletor: O coletor é responsável por capturar descritores do tráfego de rede, e normalmente está conectada em algum ponto de interconexão da rede, como por exemplo: roteador, bridge ou firewall. O desempenho do coletor depende dos equipamentos da rede utilizados para a coleta, principalmente em redes de grande tráfego. Alguns firewalls também atuam como coletores, armazenando informações para os NIDS (NORTHCUTT; NOVAK, 2002 apud PERLIN, 2010).
- Analisador: é um componente responsável por verificar os dados anteriormente coletados buscando por eventos que indiquem uma intrusão ocorrida ou que esteja em andamento, tendo diferentes abordagens para análise dos dados, como a baseada em assinaturas ou anomalias ambas descritas no decorrer do trabalho.
- Banco de dados: é onde ficam guardadas as informações dos NIDS e os eventos suspeitos, para que posteriormente possam ser buscadas e analisadas de forma mais detalhada.
- Notificador: é o sistema responsável pelo envio de alertas ao administrador ou responsável designado. Ocasionalmente ocorrem alertas falsos positivos, sua frequência pode indicar um sistema com pouca confiança.
- Atuador: é uma ferramenta que é presente somente nos IPSs portanto não faz parte deste trabalho. Ela possui a capacidade de execução de ações automatizadas quando uma suspeita é detectada pelo IDS, basicamente tem o objetivo de fazer ligação entre o administrador do sistema e o sistema de detecção, sendo o monitor usado para configurar, verificar o funcionamento do IDS e até mesmo envio de alertas. Esta peça porém não será alvo de estudo deste trabalho.

2.3 Posicionamento de um IDS na rede

Em alguns casos a análise feita pelos NIDS abrange segmentos da rede mais sensíveis, dependendo da política de segurança, devido a este fato normalmente esses sistemas ficam após os firewalls e roteadores, conforme é apresentado por REHMAN (2013) na figura 2:

Figura 2 – Posicionamento de um IDS segundo U. R Rehman, (2013)



http://ptgmedia.pearsoncmg.com/images/0131407333/downloads/0131407333.pdf

O posicionamento dos IDS depende de uma série de fatores, esses fazendo referência à topologia da rede e as atividades de intrusão que se deseja monitorar. Os ataques têm a sua origem em qualquer ponto da rede, tanto em redes locais ou outras. Muitas vezes, pode não ser suficiente a colocação dos equipamentos de monitoramento apenas nos pontos de entrada na rede (M., 2006).

2.4 Alertas e interferências (ruídos)

Os Sistemas de detecção de intrusão são normalmente baseados em regras pré-estabelecidas para mapear comportamentos anômalos na rede em determinados hosts que devem ser monitorados pelos mesmos.

Porém da mesma forma que os alertas por regras são necessários para os administradores, eles podem ser falsos alertas ou ruídos da rede. Fica a cargo de quem monitora a rede saber interpretá-los e filtrá-los para que somente alertas relacionados a ataques reais sejam notificados, pois do contrário eles podem causar um esforço desnecessário (investigação) de recursos e pessoas, além de causar a perda de credibilidade da ferramenta com relação aos responsáveis.

Na figura 3 está retratado um exemplo de ruído, sabendo que esta é uma rede controlada e que no tempo em que é demonstrado, não houve qualquer tipo de ataque ou tentativa de comportamento malicioso. Neste caso, os ips 10.0.0.12 e 10.0.0.15 são monitorados pelo IDS.

Figura 3 – Exemplo de ruído em uma rede

Autor

2.5 Snort

O Snort é uma ferramenta de código aberto de detecção de intrusão de rede (NIDS) disponível gratuitamente. É essencialmente um IDS baseado em regras (conjunto de assinaturas), no entanto existem plug-ins para detectarem anomalias nos cabeçalhos de protocolo. As regras são armazenadas em ficheiros e podem ser modificados por um editor de texto simples. Os ficheiros de regras são referenciados no ficheiro de configuração snort.conf. No momento em que a aplicação inicia, são criadas as respectivas estruturas de dados internas, que irão aplicar as regras aos dados capturados.

Quanto mais regras forem utilizadas, maior será a carga para processar os dados em tempo real. Desse modo é importante implementar o maior número de assinaturas no menor número de regras possíveis.

O Snort já vem com um conjunto vasto de regras para detectar atividades de intrusão. A esse conjunto podem-se adicionar regras próprias ou remover algumas das pré-definidas. Ele também é capaz de efetuar a análise de protocolos e pesquisa de conteúdos de forma a poder detectar uma variedade de ataques tais como:

- Buffer overflows
- Stealth port scans

- Common Gateway Interface (CGI) attacks
- Server Message Block (SMB) probes
- Operating system fingerprinting attempts

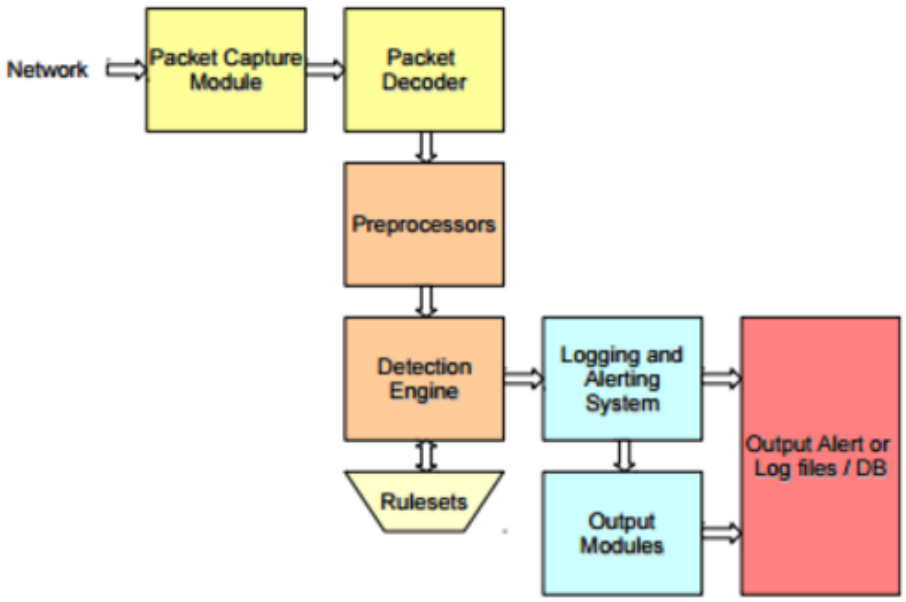
O Snort pode ser configurado em três modos:

- Modo Sniffer: apenas lê os pacotes da rede mostrando-os de forma contínua no console.
- Modo packet logger: é semelhante ao anterior, com a diferença de redirecionar o output para disco.
- Modo network intrusion detection: permite a análise do tráfego da rede tentando encontrar algum padrão descrito nas regras previamente estabelecidas e atuar em conformidade (alertas).

Adicionalmente às regras fornecidas pelo Snort, poderão ser implementadas outras regras para fazer face a requisitos específicos de um determinado ambiente. Existem também comunidades online onde especialistas em detecção de intrusão compartilham as suas regras para a detecção de novos tipos de ataque.

A figura 4 a seguir representa os componentes do Snort:

Figura 4 – Componentes do Snort



http://hacktracking.blogspot.com.br/2014/08/snort-rule-structure-and-syntax.html

1) **Packet Capture Module –** Neste módulo são recolhidos os pacotes da rede através de bibliotecas internas do próprio Snort.

2) **Packet Decoder –** O decodificador de pacote ajusta os pacotes capturados em estruturas de dados, identificando qual o protocolo em uso por um determinado pacote e compara os dados com as regras autorizadas para aquele protocolo.

3) **Preprocessors –** Os pré-processadores são componentes ou plug-ins que podem ser utilizados com o Snort para organizar ou modificar os pacotes de dados antes de o motor de detecção (Detector Engine).

4) **Detection Engine –** Este é o componente mais importante do Snort e tem como responsabilidade detectar a existência de alguma atividade de suspeita em um determinado pacote. As ações envolvem registro(logging) ou geração de alerta.

5) **Logging and Alerting System –** Se algum padrão corresponder a uma regra do mecanismo de detecção, será criado um alerta. Esse alerta poderá ser gravado num ficheiro de log ou mesmo em um banco de dados.

6) **Output Modules –** Os módulos de saída são plug-ins utilizados para enviar os alertas via popup em sistemas Windows ou via socket no caso de UNIX. Os alertas podem também ser enviados para uma base de dados.

2.5.1 Snort rules

Uma das melhores características do Snort é o seu módulo de regras e as regras em sí. O módulo de regras disponibiliza uma linguagem relativamente extensa, e permite com que os próprios usuários possam escrever as suas próprias regras, afim de que estas se adaptem ao ambiente em que o Snort está alocado.

Uma regra pode ser quebrada em duas partes principais: O cabeçalho e as opções. O cabeçalho contém a ação que deve ser tomada, o protocolo em que a regra se aplica e os endereços e portas de destino e de origem.

As opções permitem o usuário criar uma descrição para a mensagem de alerta e também verificar uma grande variedade de atributos dos pacotes. Abaixo esta a forma geral de uma regra do Snort:

action proto src_ip src_port direction dst_ip dst_port (options)

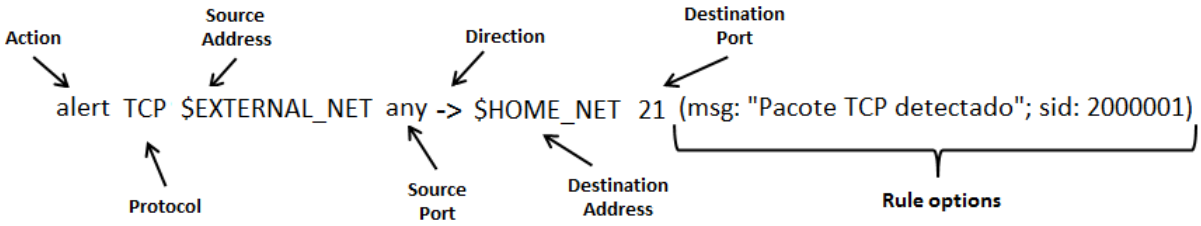
Onde as partes específicas são definidas como:

- **action** Esta parte especifica que ação deve ser tomada quando for acionada pelo evento de intrusão. As opções são “alert” para alertar ou “pass” para ignorar.

- **proto** Determina qual protocolo deve ser monitorado. Os protocolos suportados são TCP, IP, UDP e ICMP.
- **src_ip** (Source Address) IP de origem do ataque. Normalmente são usadas variáveis para cobrir um número viável de IPs e fazer a regra ainda ser legível. Para monitorar ataques de qualquer IP, pode-se utilizar a palavra reservada "any". Mas por padrão, existe a variável "$EXTERNAL_NET" que é utilizada pelo Snort para determinar os IPs dos quais os ataques devem ser monitorados, sendo que, por uma boa prática e para diminuir falsos alertas, "$EXTERNAL_NET" é instanciada como sendo "any !$HOME_NET", ou seja, todos os IPs menos os considerados IPs locais.
- **src_port** - (Source port) Porta da qual o pacote deve ter a origem para ser válida. Da mesma forma como o src_ip, este parâmetro tem a opção "any" para deixar livre e ampliar o monitoramento para qualquer porta, além de suportar o uso de uma variável. Fora isso, podem ser utilizados listas de portas ou intervalos (65-443).
- **direction** Determina a direção do tráfego da rede, podendo ser:
  - De dentro para fora "->"
  - De fora para dentro "<-"
  - Bi-direcional "<>"
- **dst_ip** (Destination Address) IP de destino do pacote. Este parâmetro segue as mesmas regras do src_ip e é responsável por determinar o IP (ou IPs) que o pacote está endereçado.
- **dst_port** (Destination port) Porta de destino. Segue as mesmas regras do src_port sobre delimitação de quais devem ser monitoradas pelo Snort, porém esta determina para qual porta o pacote está sendo enviado.
- **(options)** Este é o corpo da regra. Nesta parte é possível especificar maiores detalhes de quando a regra deve ser aplicada, como determinando o conteúdo do pacote inspecionado, a forma como a regra deve ser notificada (mensagem a ser exibida) e também a categoria do alerta.

Na figura 5 está um exemplo de regra simples:

Figura 5 – Exemplo de regra simples



Autor

Onde suas partes podem ser definidas a partir de:

- **Action:** “alert” vai alertar, da forma como o Snort estiver configurado, caso a regra seja violada.
- **Protocol:** “TCP” somente tráfego do protocolo TCP irão disparar esta regra.
- **Source Address:** “$EXTERNAL_NET” todos os endereços configurados como externos na configuração do Snort.
- **Source port:** “any” qualquer porta.
- **Direction:** “->” De fora para dentro da rede monitorada.
- **Destination Address:** “$HOME_NET” todos os endereços configurados como internos na configuração do Snort.
- **Destination Port:** “21” Somente pacotes endereçados a porta 21 ativarão esta regra.
- **Rule options:** msg: “Pacote TCP detectado” mensagem que será exibida pelo Snort. sid: 2000001 número de identificação do evento. Utilizado em conjunto com outros arquivos de configuração para classificação do alerta, como o seu nível de prioridade.r

2.5.2 Instalação do Snort

Primeiramente, é necessário instalar algumas dependências:

```
sudo apt-get install -y build-essential libpcap-dev libpcre3-
    dev libdumbnet-dev bison flex zlib1g-dev
```

Em seguida, deve-se instalar o DAQ: dependência responsável por obter os dados do Snort e disponibilizar para análise de uma forma unificada, o que facilita exportação para um maior número de analisadores.

```
mkdir ~/snort_source
cd ~/snort_source
wget https://www.snort.org/downloads/snort/daq-2.0.5.tar.gz
tar -xvzf daq-2.0.5.tar.gz
cd daq-2.0.5
./configure
make
sudo make install
```

Após as dependências, a instalação do Snort segue a mesma lógica anterior: baixa-se o arquivo fonte do site oficial do Snort, e instála-se o no diretório local do linux de preferência.

```
cd ~/snort_source
wget https://www.snort.org/downloads/snort/snort-2.9.7.3.tar.gz
tar -xvzf snort-2.9.7.3.tar.gz
cd snort-2.9.7.3
./configure  enable  -sourcefire
make
sudo make install
sudo ldconfig
sudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

Seguindo estes passos, a instalação já deve estar concluída e o Snort já vai estar pronto para receber regras novas e ser posto em funcionamento. Afim de testar o sucesso da instalação, pode ser utilizado o comando abaixo:

```
/usr/sbin/snort -V
```

Deve ser exibida uma mensagem parecida com a da imagem 6:

Figura 6 – Mensagem de sucesso na instalação do Snort

https://itfellover.com/category/building-an-ethical-hacking-lab

3 Nmap

Liberado para o público em setembro de 1997, o Nmap segundo seu criador e autor do livro base para este trabalho:

> Meu objetivo era consolidar o fragmentado campo de scanners de portas de propósitos especiais em uma ferramenta livre, poderosa e flexível, fornecendo uma interface consistente e uma implementação eficiente em todas as técnicas práticas de exame de portas.(LYON, 2009)

O Nmap é um utilitário livre e de código aberto que, tendo o apoio de seu desenvolvimento na comunidade, cresceu até se tornar o scanner de segurança de redes mais popular do mundo[1]. Com sua exploração baseada em pacotes de IP, o Nmap consegue:

- Determinar quais hospedeiros estão disponíveis.
- Detectar SO - Remotamente determina o sistema operacional e características do hardware do alvo.
- Executar exame ocioso.
- Interação com o alvo, através de scripts.
- Examinar portas - Detecção da versão/serviço utilizando determinada porta.
- Quais tipos de filtros de pacotes/firewall estão em uso entre outras funcionalidades.

Além disso, o Nmap vem disponível em duas distribuições, a CUI via console e a GUI via interface gráfica através do Zenmap.

3.1 Zenmap

O Zenmap é a interface gráfica oficial para o Nmap, sendo de código aberto e multi-plataforma. Sua principal missão é facilitar o uso do Nmap por iniciantes ao mesmo tempo que proporciona acesso as opções avançadas para usuários experientes. Todos os scans podem ser salvos para análise e comparação porsterior ao exame.

A interface é bastante simples, o preenchimento básico de um exame é composto por três campos, exemplificado pela imagem 7.

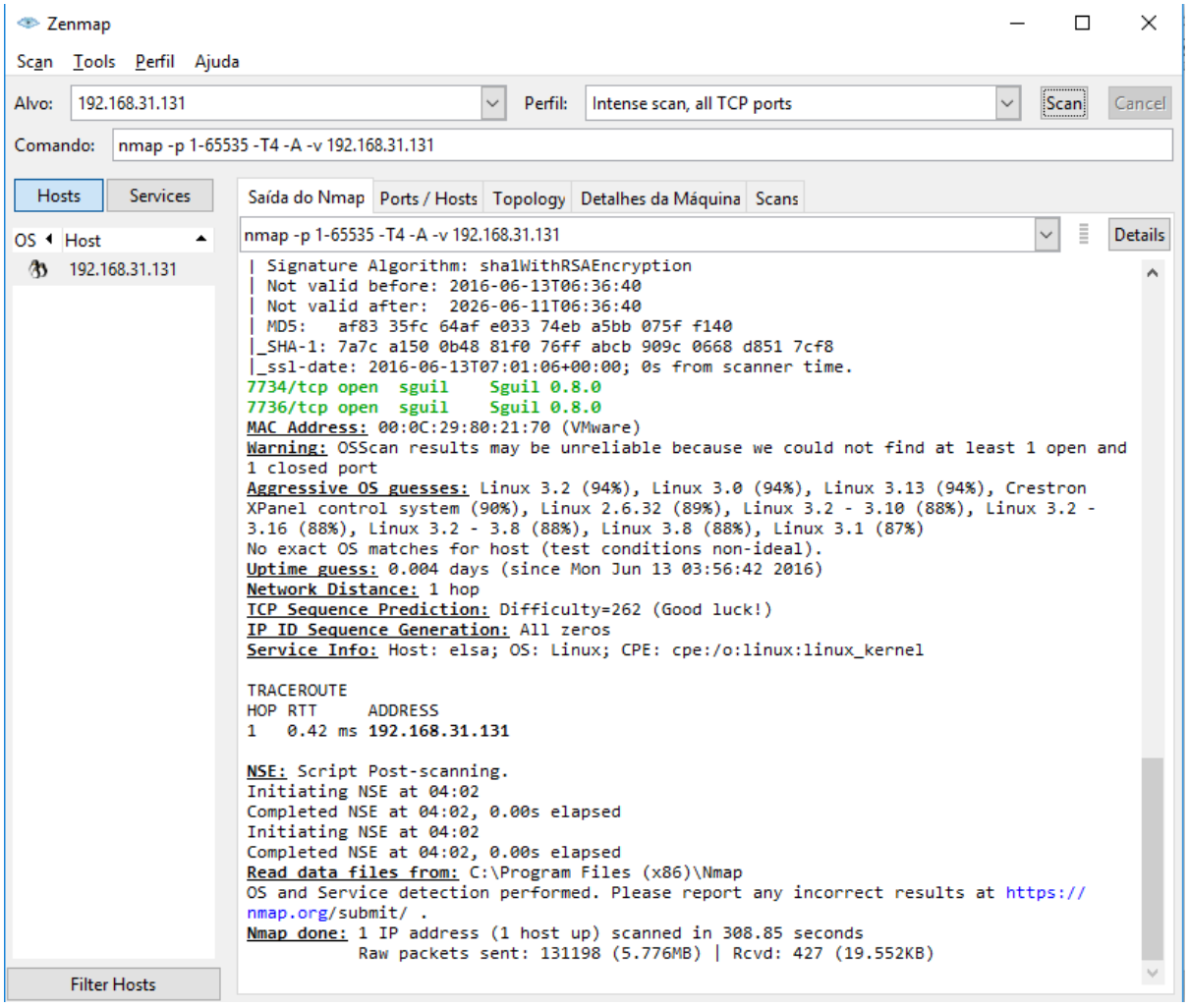
[1] Com base na frequência de download, número de alcances no Google e na parada de “popularidade” de software do Freshmmeal.net. (Dados de 2009)

1) Alvo - Host que será examinado, pode ser um IP ou um endereço DNS

2) Perfil - Exames comuns pré-configurados

3) Comando - Campo onde o comando do exame é descrito na íntegra

Figura 7 – Tela inicial do Zenmap



Autor

3.2 Exame de portas

Apesar de ter crescido bastante em funcionalidades ao longo dos seus anos, devido ao seu desenvolvimento pela comunidade, a sua essência e utilidade principal continua sendo o exame de portas. Por exemplo, o comando “nmap <alvo>” vai escanear 1000 portas TCP do alvo, classificando-as de acordo com o seu estado, sendo os possíveis: open, closed, filtered, unfiltered, open|filtered, closed|filtered.

3.2.1 O que é uma porta

Por definição, a exemplo do que o IP representa para uma máquina em uma rede, a porta é um número que representa uma abstração de software, usada para distinguir canais de comunicação (transmissão de dados). Mais detalhadamente, as portas identificam aplicações específicas em uma única máquina, sendo que as mais comuns são as portas protocoladas TCP e UDP, as quais são usadas para comunicação entre a máquina e a internet.

Por exemplo, ao se conectar a internet através de um navegador, automaticamente ele utilizará a porta TCP 80 em URLs HTTP. Já no caso de a URL utilizar o protocolo seguro HTTPS, a porta utilizada (por padrão) é a 443.

Como a maioria dos serviços está registrada em um número de portas bastante conhecido, qualquer um pode deduzir quais serviços as eventuais portas abertas representam. Nesse caso, o Nmap faz esse papel através de um arquivo nmap-services, que possui o mapeamento entre portas e serviços registrados na IANA e também outras aplicações comuns.

A IANA possui um esquema de classificação de portas que divide o espaço em três classes:

1) Portas bem conhecidas: Estas são portas reservadas. Estão dentro da faixa de 1 a 1023 e que foram registradas na IANA para um determinado serviço. Ex: 22, 25 e 80 para os serviços SSH, SMTP e HTTP respectivamente.

2) Portas registradas: Portas que também foram registradas na IANA e inclusive grande parte é tão utilizada quanto as portas reservadas. A diferença entre estas portas e as “portas bem conhecidas”, além da sua faixa se estender entre 1024 e 49151, é de que usuários sem privilégios de administrados podem se conectar a elas e rodar seus serviços.

3) Portas dinâmicas e/ou privadas: Esta faixa de portas, de 49152 até 65535, é reservada pela IANA para a distribuição sob demanda de aplicações normalmente quando estas não especificam a porta desejada (porta 0). Muitas vezes é utilizado para transferência de arquivos, como por clientes FTP ao solicitar uma transferência de modo ativo ou clientes P2P e de mensagens instantâneas.

3.2.2 Exame de portas

Estes estados mostram a forma como o nmap as enxerga, não necessariamente o seu estado real. Por exemplo, se for feito um exame da porta 135/TCP pelo próprio host, possivelmente está será apresentada como open. Porém se o mesmo exame for feito por outro host de fora da rede local e através de um firewall, muito possivelmente

a mesma porta será exibida como filtered. Abaixo o detalhamento de cada estado possível de uma porta segundo o nmap.

1) **open:** Uma aplicação está ativamente aceitando ou conexões TCP, ou datagramas UDP ou associações SCTP. Este é o estado mais visado e desejado pelos atacantes pois, além de exibir os serviços que estão em uso, são o caminho de entrada para explorar o host. Na prática é como se ao enviar um pacote SYN, o nmap recebesse a resposta SYN/ACK

2) **closed:** A porta está acessível (está recebendo e respondendo os pacotes do nmap) mas não existe uma aplicação ouvindo-a. Por um lado, este estado não é tão útil como o open, mas um atacante mais perseverante pode utilizar esta porta e fazê-la útil mesmo assim. Ela pode comprovar que um host existe e está conectado a uma rede, pode também ser utilizada para descobrir qual SO está sendo utilizado e até após um certo tempo pode se tornar open. Na prática é como se ao enviar um pacote SYN, o nmap recebesse a resposta RST

3) **filtered:** Não foi possível determinar se a porta está open devido ao fato de existir um filtro entre o atacante e o alvo. Esta situação pode ser causada tanto por Firewalls no caminho do ataque, firewall no próprio host alvo ou regras de roteamento. Na prática é como se ao enviar um pacote SYN, o nmap não recebesse nenhuma resposta.

4) **unfiltered:** A porta está acessível, porém não foi possível determinar se ela está open ou closed. Este estado só é gerado pelo exame ACK, o qual é utilizado para mapear o conjunto de regras de um firewall. É possível fazer um refinamento deste resultado fazendo um exame SYN ou FYN e talvez conseguir o estado real.

5) **open | filtered:** Estado reservado para as portas as quais o Nmap não consegue determinar se estão abertas (open) ou filtradas (filtered). Isto ocorre em exames em que portas abertas não enviam resposta, como no exame de Natal (Xmas Scan).

6) **closed | filtered:** Parecido com o estado anterior, este é resevado para portas que o Nmap não consegue diferenciar se estão no estado fechada (closed) ou filtradas (filtered)

3.3 Criando exames de portas/vulnerabilidade com Nmap

Uma parte fundamental no sucesso de um exame de porta ou de vulnerabilidade, é saber como elaborá-lo da melhor forma possível através da sintática permitida pelo Nmap. É possível encontrar diversos exemplos prontos na internet, e alguns ataques

"enlatados" dentro do próprio Zenmap, mas o diferencial para driblar ou ludibriar qualquer forma de detecção é saber manipular os ataques básicos através dos parâmetros encontrados em (LYON,).

O comando mais simples do Nmap é "nmap <alvo>", sendo que o alvo pode ser o nome de um hospedeiro ou diretamente o endereço IP (v4 ou v6). Suas ações são as seguintes:

1) Pinga o alvo, por omissão com um pacote de requisição de eco de ICMP e um pacote TCP ACK para a porta 80 para que assim, possa determinar se o host está no ar. Esta etapa pode ser desabilitada através do parâmetro -PN.

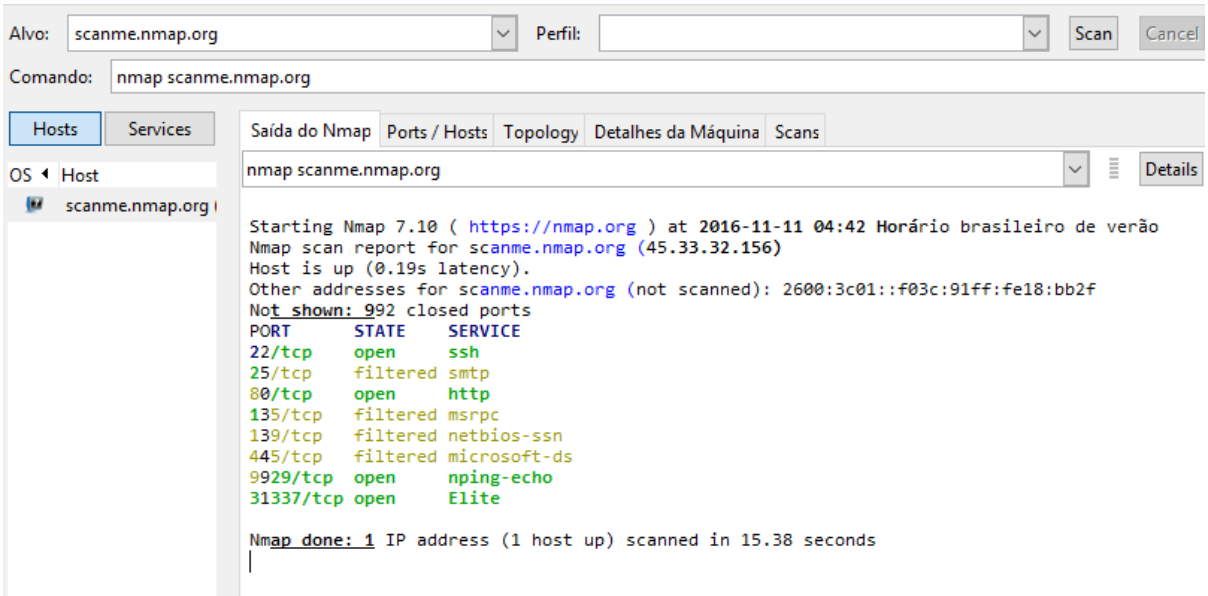
2) Converte o endereço IP do alvo de volta ao nome, utilizando uma consulta de DNS reverso. Para pular esta etapa, pode-se incluir o parâmetro -n.

3) Lança um exame de portas TCP nas 1000 portas mais populares listadas no arquivo nmap-services. Por via de regra é utilizado o exame via SYN, mas pode ser usado o exame por connect caso o usuário não tenha os privilégios necessários. Esta mudança é feita automaticamente.

4) Exibe os resultados na saída padrão de forma legível a humanos.

Na figura 8 está um exemplo de saída do exame padrao do Nmap no Zenmap:

Figura 8 – Exemplo de scan padrão via Zenmap



Autor

A partir deste ataque padrão, é possível utilizar parâmetros para incluir, retirar ou alterar a forma e o objetivo do ataque. A seguir, seguem algumas das técnicas possíveis.

3.3.1 Técnicas de exame de portas

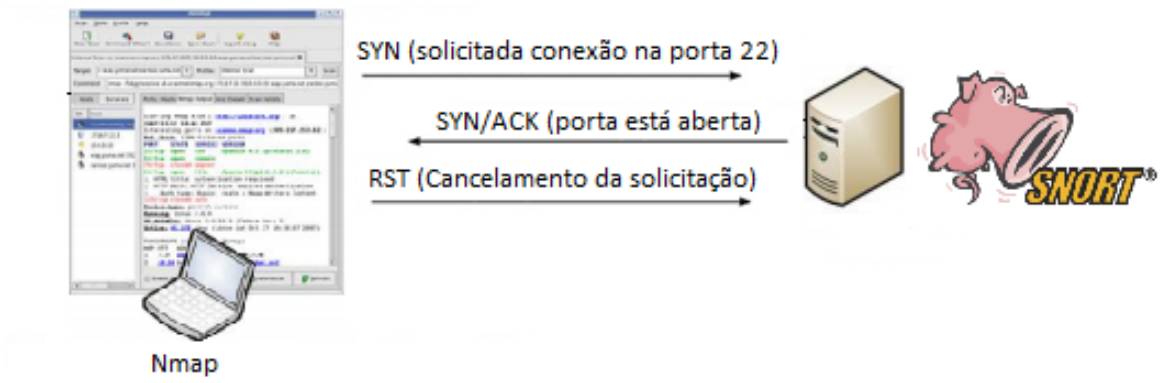
Scannear uma porta consiste em uma série de mensagens enviadas a partir de um host (atacante) na tentativa de obter informações referentes a quais serviços cada uma das portas “conhecidas” o host alvo possui. Sabendo-se que as portas são os lugares de onde as informações entram e saem de um computador saber quais delas estão abertas pode ser a abertura necessária para um atacante executar suas ações.

Existem vários tipos de scans de redes, dentre os quais o Nmap disponibiliza os seguintes:

3.3.1.1 De TCP por SYN (-sS)

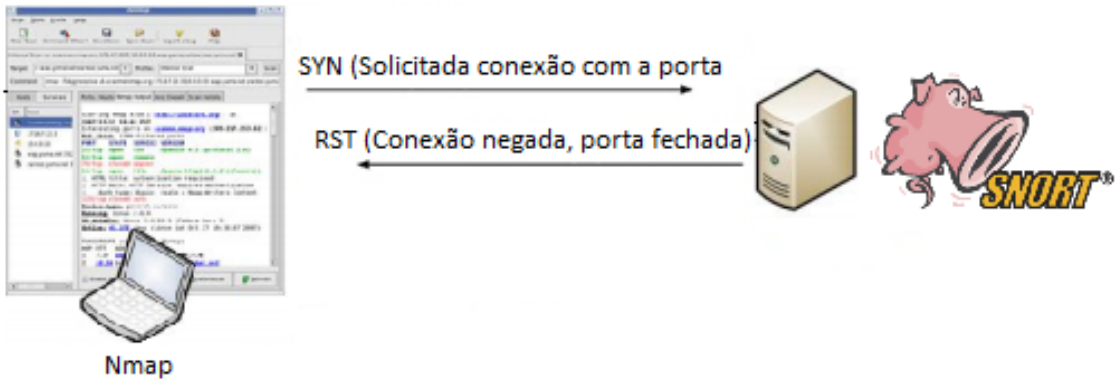
Por parte no Nmap, este é o exame mais popular, por ser a forma mais rápida para examinar as portas do protocolo mais comum (TCP). Costuma receber o apelido de “Invisível” (Stealth). Funciona basicamente com o Nmap enviando um pacote TCP com o sinalizador SYN ligado (primeiro passo da saudação em três fases do TCP). E então dependendo da resposta, ou falta dela, o Nmap determina o estado da porta.

Figura 9 – Scan SYN em porta aberta



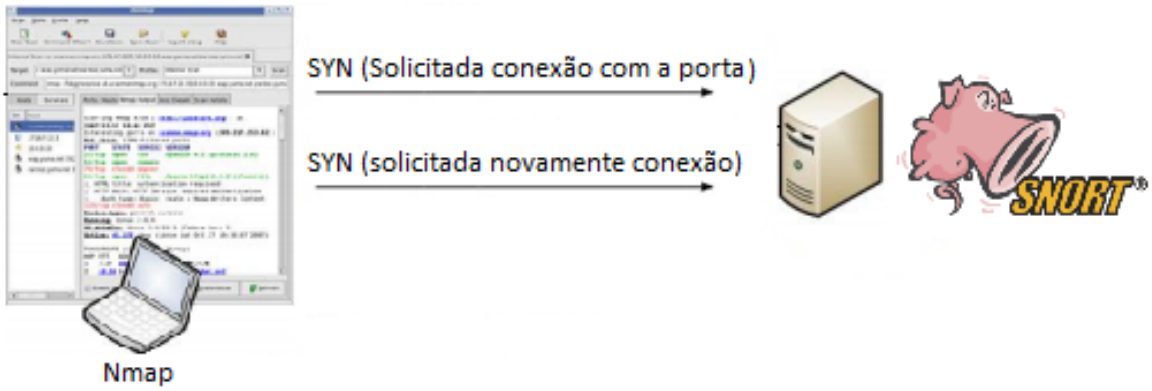
Autor

Figura 10 – Scan SYN em porta fechada



Autor

Figura 11 – Scan SYN em porta filtrada

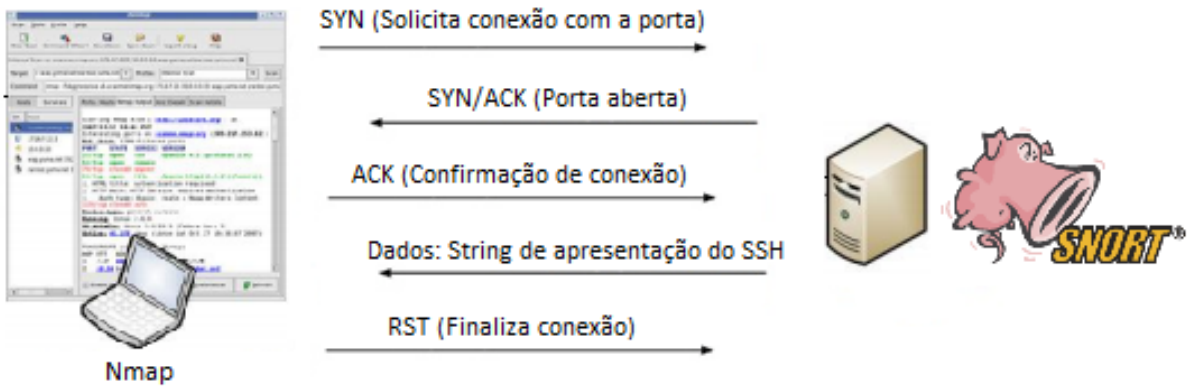


Autor

3.3.1.2 TCP por connect (-sT)

Normalmente utilizado em substituição da técnica de TCP por SYN, é empregada nos casos em que os usuários não possuem privilégios ou quando os alvos estão identificados somente sob o IPv6. Ao invés de escrever pacotes crus, e de terminar a conexão caso receba a resposta SYN/ACK, o Nmap solicitada ao sistema operacional que estabeleça uma conexão com a máquina alvo.

Figura 12 – Exame por connect em porta aberta



Autor

3.3.1.3 UDP (-sU)

Este é o único exame que pode ser combinado com os outros disponibilizados pelo Nmap. Ele ativa o scan das portas que respondem sob o protocolo UDP, como a 53 (DNS), 161/162 (SNMP) e 67/68 (DHCP).

Respostas à prova	**Estado atribuído**
Qualquer resposta UDP da porta alvo	open (aberta)
Nenhuma resposta recebida, mesmo depois de retransmissões	open|filtered (aberta ou filtrada)
Erro de porta inalcançável de ICMP (código 3)	closed (fechada)
Outros erros de porta inalcançável	filtered (filtrada)

3.3.1.4 TCP por FIN, de Natal (Xmas) e nulo (-sF, -sX, -sN)

Estes três exames seguem o mesmo comportamento, diferem somente nos bits sinalizadores ligados: Nulo não liga nenhum bit, o FIN liga somente o bit FIN e o Natal (Xmas) liga os sinalizadores FIN, PSH e URG. Todos exploram a mesma brecha encontrada na RFC 793 do TCP, para definir se uma porta está aberta ou fechada, onde é definido que:

> se a porta [de destino] estiver CLOSED (fechada) . . . um segmento de chegada que não contenha um RST fará com que um RST seja enviado como resposta

Portanto, ao examinar sistemas que se adequa, a este texto de RFC, qualquer pacote que não contenha os sinalizadores SYN, RST ou ACK ligados irá resultar em um RST sendo retornado caso a porta esteja fechada, e nenhuma resposta caso a mesma esteja aberta ou filtrada. Abaixo uma tabela com os comportamentos esperados:

Respostas à prova	**Estado atribuído**
Nenhuma resposta recebida, mesmo depois de retransmissões	open|filtered (aberta ou filtrada)
Pacote TCP RST	closed (fechada)
Erro de inalcançável de ICMP	filtered (filtrada)

A principal vantagem deste tipo de scan é de que eles tem mais probabilidade de atravessar firewalls e roteadores com filtragem de pacotes. Além disso têm chance de serem mais discretos do que o exame por SYN. O problema é que muitas das vezes não é possível distinguir as portas abertas das filtradas, e também dependem de que o sistema alvo aplique a RFC 793.

3.3.1.5 TCP por ACK (-sA)

Este é um scan que serve mais como apoio para determinar informações extras sobre um sistema. É geralmente utilizado para mapear conjuntos de regras de firewalls, determinando se os mesmos são de estado ou não. No caso de uma porta estar aberta ou fechada, o Nmap interpreta a resposta como “unfiltered”, ou seja, não foi filtrada por um firewall e foi possível estabelecer algum estado.

Um exemplo de uso do scan por ACK em combinação para determinar o estado de uma porta, é casar o seu resultado com o resultado do exame de TCP por FIN. Neste cenário, se uma porta é determinada como open|filtered pelo scan TCP de FIN, e unfiltered no scan de TCP por ACK, pode-se deduzir que a porta então está aberta “open”. Pois fazendo uma tabela verdade, de um lado ela esta aberta ou filtrada e de outro aberta ou fechada, resultando no único estado comum: aberta.

3.3.1.6 TCP ocioso (-sI <hospedeiro>)

Em 1998 um pesquisador de segurança chamado Antierez desenvolveu uma técnica que permitiria exames de portas de forma não fosse possível para o alvo descobrir diretamente de onde estariam vindo os verdadeiros ataques, ao mesmo tempo que obteria resultados precisos sobre as portas de TCP. Esta técnica ficou conhecida como exame ocioso, pois o atacante forja sua identidade de forma que pareça que uma máquina zumbi inocente fez o exame.

Este tipo de exame se baseia em três princípios básicos do comportamento de máquinas em redes:

- Ao enviar um pacote SYN (estabelecimento de sessão) a uma determinada porta a máquina responsável responderá com um pacote SYN/ACK (requisição de sessão reconhecida) caso a porta esteja aberta ou RST (resetar) se a porta estiver fechada. Com isso é possível determinar se uma porta TCP está aberta.

- Ao receber um pacote SYN/ACK que não foi solicitado por ela responderá com outro pacote RST. Caso receba um RST não solicitado ele simplesmente será ignorado.
- Todo o pacote IP tem embutido um número de identificação. Como este número na grande maioria das vezes é sequencial e incrementado de acordo com o número de pacotes que a máquina envia, é possível determinar quantos pacotes foram enviados desde a última prova feita pelo atacante.

3.3.1.6.1 Passo a passo

Fundamentalmente, após conseguir uma máquina zumbi, o exame ocioso é feito por três passos repetidos para cada porta alvo, sendo eles:

1) Determinar o ID de IP da máquina zumbi
2) Forjar um pacote SYN a partir da máquina zumbi e enviá-lo ao alvo na determinada porta. A reação do alvo a partir do recebimento do pacote fará com que o ID de IP do zumbi seja incrementado dependendo do estado da porta alvo.
3) O atacante deve provar novamente o ID de IP do zumbi. O estado da porta será determinado a partir da comparação do ID de IP obtido no passo 1.

Ao final do terceiro passo, o ID de IP da máquina zumbi terá sido incrementado em um ou dois.

No caso de ter sido incrementado em um significa que o zumbi não enviou nenhum pacote a não ser a própria resposta à prova inicial do atacante (passo 1) e isso determina que a porta não está aberta.

Já se o ID de IP tenha incrementado em dois significa que o zumbi enviou um pacote entre as duas provas feitas pelo atacante (passos 1 e 3). Este pacote extra significa que a máquina alvo enviou um pacote SYN/ACK ao zumbi em resposta ao pacote SYN forjado pelo atacante (passo 2) e portanto prova que a porta alvo está aberta.

Em casos de incremento maiores do que dois podem significar que a máquina zumbi não possui incremento sequencial de ID de IP ou então que ela pode já estar em comunicação não relacionada ao exame ocioso.

3.3.1.7 Protocolos IP (-sO)

Scans do Protocolo IP permitem que você determine quais protocolos IP (TCP, ICMP, IGMP, etc.) são suportados pelas máquina-alvo.

O scan de protocolo funciona de uma forma similar a um scan UDP. Ao invés de ficar repetindo alternando o campo de número de porta de um pacote UDP, ele envia cabeçalhos de pacote IP e faz a repetição alternando o campo de protocolo IP de 8 bits. Os cabeçalhos normalmente estão vazios, e é baseado nessa caracteristica padrão que o Snort se baseia para detectar este tipo de scan.

3.3.2 Opções

Opções complementam opcionalmente um ataque. Abaixo uma tabela com algumas das configurações mais comuns:

Opções de Scan	**Título**	**Função**
-g	Especificar porta	Usar um porta específica para enviar pacotes.
-S	Endereço IP de origem	Simula um IP ou diz para o Nmap qual IP usar.
–spoof_mac	Mac Falso	Cria um mac falso para enviar pacotes.
-p	Especificar faixa de portas	Determina quais portas escanear.
-R	Lookup reverso	Força lookup reverso.
-N	Resolução DNS	Lookup reverso rápido.
-n	Não resolução DNS	Não lookup reverso.
-sV	Versão de serviço	Examina versões de software

4 Detecção de Sistemas de Detecção de Intrusão

O IDS mais discreto é aquele que analisa o tráfego de uma rede sem nunca transmitir, existem até recursos que podem ser utilizados para que não sejam transmitidas informações mesmo que o IDS seja comprometido por atacantes pois é a partir de transmissões que os atacantes conseguem ter informações sobre a existência e detalhes de um sistema de detecção de intrusão que ocasionalmente esteja monitorando a rede visada. Porém para fornecer mais informações sobre os alertas, os IDS iniciam provas, as quais abrem espaço para que possam ser vistas pelos usuários maliciosos.

Neste capítulo serão apresentadas definições sobre detecção de IDSs em uma rede, como também as formas mais comumente utilizadas para tal e a forma como podem ser adotadas por eventuais atacantes.

4.1 Provas reversas

Uma destas provas que é muitas vezes executada pelos IDSs é a consulta de DNS reverso do endereço IP do suposto atacante, com o intuito de trazer informações mais úteis para um possível alerta tal como o nome do domínio do atacante. Porém, em muitos dos casos os usuários maliciosos controlam seu próprio rDNS, isso lhes da a chance de controlar os registros e perceber quando foram detectados, podendo com isso fornecer informações forjadas propositalmente para despistar possíveis alertas.

Ao detectarem uma atividade suspeita, alguns IDSs ainda podem ser mais intrusivos na procura de informações sobre seus atacantes. Esta pesquisa é baseada na solicitação de informações de NetBIOS do Windows de volta ao atacante.

Sabendo disso, é possível que um atacante tire vantagem deste procedimento padrão. Tendo em mãos uma ferramenta simples de envio de pacotes, pode-se enviar para a rede visada um pacote que por sua vez gera alerta das instancias de IDSs que estão observando a rede, que então tentariam obter informações deste ataque através de provas reversas. Partindo deste ponto o atacante pode explorar usando as brechas que serão discutidas posteriormente ao longo deste trabalho.

4.2 Súbitas mudanças no firewall e pacotes suspeitos

IDS é como são classificados os Sistemas que atuam passivamente, ou seja, somente monitoram a rede e geram alertas para os administradores. Já quando o Sistema atua ativamente, atuando em linha na rede, para que assim possam restringir o fluxo de pacotes quando perceber alguma ameaça, como quando alguém tenta examinar

suas portas. Porém é esta mesma ação de prevenção que faz com que o atacante perceba a presença de um IPS no caminho da rede, como quando após escanear portas reportadas como abertas elas subitamente se tornam fechadas.

Outra dica que os IPS da inadvertidamente para os atacantes é quando ao iniciar um ataque ou scan, recebe como resposta pacotes suspeitos, os IPSs tomam essa ação numa tentativa de derrubar as conexões, e a forma de detecção destes pacotes forjados será discutida ao longo deste trabalho.

4.3 Convenções de nomeação

Todo o sistema tem seus nomes padrões, de identificação de si mesmo e de variáveis. Partindo deste princípio, um atacante que tenha em mãos um programa capaz de examinar alvos retornando nomes de hospedeiros como o Nmap, pode obter uma lista que descreve quais sistemas estão rodando, inclusive a presença de um IPS.

Porém esta é uma faca de dois gumes, pois da mesma forma que o DNS ao mostrar um hospedeiro sendo “realsecure” e o atacante pode deduzir que é um IPS, ele pode deduzir que “bugzilla.securitufocus.com” é uma instancia do Bugzilla(software de acompanhamento de bugs) rodando, mas ao examinar o host especificamente é possível descobrir que na verdade ele é uma instancia do firewall da Symantec Raptor.

4.4 Saltos de TTL inexplicados

Falhas inexplicadas ou maquinas suspeitas ao traçar a rota até o alvo é mais uma forma de se detectar certos IDSs. Para faze-lo existem comandos aclopados no caso do windows pode ser usado o “tracert”, mas para uma sondagem mais eficiente o Nmap possui rotinas específicas. Esta técnica só é eficiente em IDSs que atuam em linha na rede, diferente dos que somente agem passivamente sem fazer parte da rota (paralelamente).

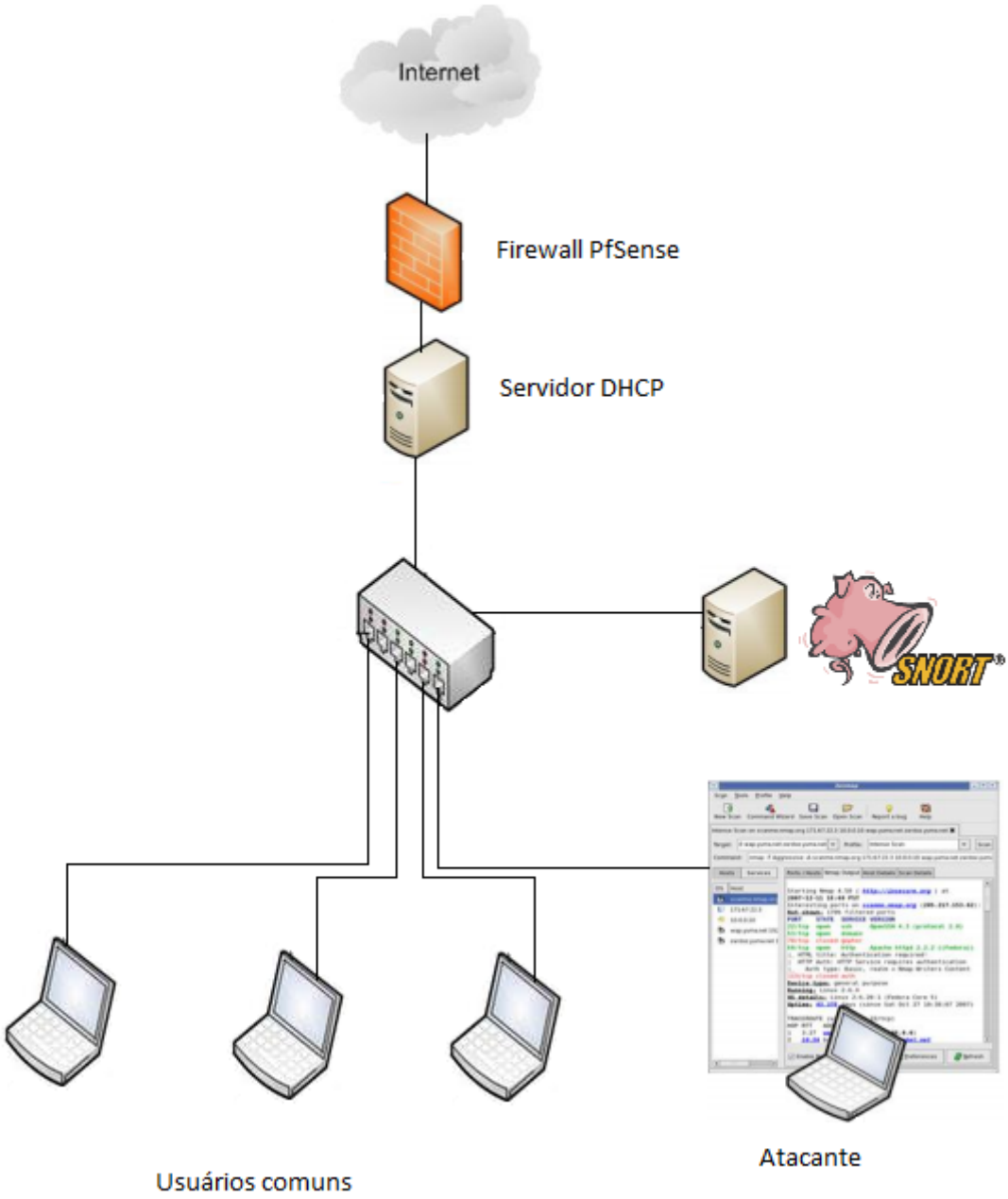
5 Ambiente de teste

Para testar a efetividade das estratégias de subversão, é necessário utilizá-las na prática, em uma rede ou um segmento de rede. Este capítulo trata da estrutura da rede utilizada para a prática dos testes de ataques e de como estes foram implementados na sua forma geral.

5.1 Estrutura da rede

Para este trabalho, foi criada uma rede a partir de maquinas virtuais que simulam uma estrutura padrão, onde existe um servidor exposto a internet, o qual serve a uma rede interna com serviços e acesso a internet. Foi utilizado o software de virtualização Oracle VM VirtualBox e imagens emuladas de sistemas operacionais que serão enumerados em seguida. A figura 13 ilustra a estrutura desta rede.

Figura 13 – Estrutura da rede



Autor

Pode-se perceber que, diferente dos casos mais comuns de ataque e diferente da maioria dos estudos disponíveis, o atacante está vindo de dentro da rede, e não da internet. Nesta simulação, o atacante tenta ganhar acesso a informações sensíveis do servidor que provê serviços e conexão para o resto da rede.

A topologia inclui um servidor responsável por prover os IPs (DHCP) e dar acesso a internet para a rede interna. Já o servidor responsável pelo monitoramento da rede está recebendo uma conexão espelhada do switch em modo promíscuo, portanto terá acesso a todo o tráfego sem interferência direta no desempenho. É importante

ressaltar que, neste tipo de topologia, o Snort age somente como sniffer, ou seja, tem o papel de monitoramento e detecção (IDS), e não tem poder de intervenção e prevenção (IPS). Finalmente existem as máquinas alocadas na rede interna, onde inclusive está o atacante. Abaixo uma lista detalhada dos elementos que compõem a rede:

- Atacante:
  - IP: 10.0.0.16
  - Sistema operacional: Linux Ubuntu 64 (64-bit)
  - Memória principal: 2048 MB
  - Disco rígido: 10 GB
  - Nmap/Zenmap 7.12
- Servidor Snort:
  - IP: 10.0.0.14
  - Sistema operacional: Linux Ubuntu 64 (64-bit)
  - Memória principal: 1024 MB
  - Disco rígido: 10 GB
  - Snort 2.9.8.3
- Servidor DHCP:
  - IP: 10.0.0.12 e 10.0.0.15, interno e externo respectivamente
  - Sistema operacional: PFSense (32-bit)
  - Memória principal: 1024 MB
  - Disco rígido: 10 GB
- Demais máquinas:
  - IP: 10.0.0.18 e acima.
  - Sistema operacional: Linux Ubuntu (64-bit)
  - Memória principal: 512 MB
  - Disco rígido: 10 GB
  - Possuem um serviço Apache HTTP/80 executando

5.2 Ataque

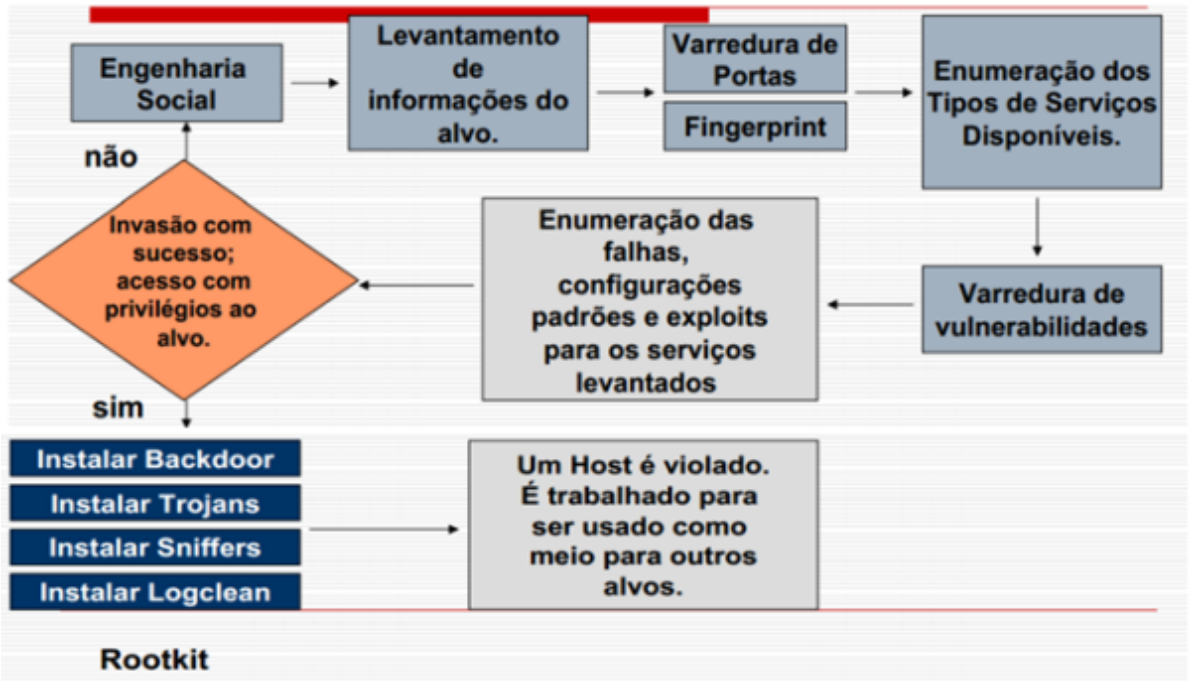
Como foco de estudo deste trabalho, a varredura de portas é a metodologia de subversão escolhida para testar o sistema de detecção de intrusão. Os ataques seguirão, de forma geral, a seguinte metodologia: um teste , com o ataque sem manipulação por parte do atacante, e em seguida o “depois” ou “manipulado”, já contando com a estratégia visada para testar o Snort.

É importante ressaltar alguns pontos:

- Os ataques estão sendo feitos de dentro da própria rede monitorada, portanto não sofrem influência externa (firewalls que não o do host por exemplo);
- Em vários resultados de testes, houve alertas que devem ser desconsiderados pois são ruídos da rede, como já explicado nos capítulos anteriores. Estes, são todos os alertas oriundos do host **10.0.0.15;**
- Os ataques partem sempre da máquina com IP **10.0.0.16**, exceto nos casos do “Exame ocioso” e do “Simulação de exames de porta”.

De forma geral, um ataque de varredura de portas (alvo deste trabalho), está inserido em um contexto mais complexo de ataque de penetração, afim de obter controle ou informações sensíveis de uma máquina. O fluxograma deste tipo de ataque está representado na figura 14:

Figura 14 – Anatomia de um teste de penetração - (MELO, 2008)



6 Evitando Sistemas de Detecção de Intrusão

Independente da companhia que desenvolveu, as tecnologias que utiliza ou em qual sistema operacional está rodando, todos os IDSs têm algo em comum: são baseados em regras. Estas regras são o fator que determina se um evento é ou não uma ameaça e qual a sua gravidade. Sabendo disso, os atacantes têm diversos métodos para se proteger durante suas explorações. Este capítulo trata da exploração invisível de portas.

6.1 Retardar as inspeções de portas

Uma das estratégias utilizadas pelos IDSs para monitoramento de portas é inspecioná-las utilizando limiares. O sistema observa um determinado número de provas em uma determinada janela de tempo. Isso é utilizado por dois motivos principais: economia de recursos (computacionais) e evitar falsos positivos pois o alerta só será gerado caso seja detectado provas de um mesmo host de origem dentro deste espaço de tempo.

Tendo em vista esta estratégia, e utilizando o Nmap, é possível traçar exames de portas abaixo dos limiares de detecção, fazendo com que a inspeção entre portas seja feita com intervalos de tempo maiores e também sem paralelismo, assim teoricamente as inspeções estariam “abaixo do radar”.

Porém, os IDSs mais modernos utilizam outra estratégia em conjunto com o limiar fixo que determina a janela de tempo de inspeção, a Janela Deslizante. Esta é muito mais robusta e inteligente do que a estratégia fixa, pois funciona da seguinte maneira: Ao receber uma prova, o IDS dispara um evento para o gerenciador de janela deslizante, que por sua vez aumenta o espaço de tempo que o sistema observará as provas vindas de determinado host. Sendo assim, o comportamento esperado utilizando a janela deslizante se mostra mais evidente a partir da segunda prova, onde ao contrário da estratégia de tempo fixo que faria o sistema ficar em alerta por **T**, neste caso faria o IDS ficar por **T + tempo extra padrão**, sendo T o tempo inicial fixo.

6.2 Experiência prática 1 - Fragmentar pacotes

No que se refere a Sistemas de detecção de intrusão, a fragmentação de pacotes pode ser um problema, particularmente por causa do tratamento de irregularidades entre eles, como a sobreposição das partes e as expirações de montagem de fragmentações são ambíguos e diferem bastante entre as plataformas. Estes fatores fazem com

que os IDSs tenham que deduzir como o sistema que recebe os pacotes vai interpretar um pacote.

6.2.1 Objetivo

Conseguir informações sensíveis da máquina alvo sem que sejam disparados alertas pelo Snort utilizando a estratégia de fragmentação de pacotes.

6.2.2 Metodologia

Este ataque será composto por três etapas: Ataque original, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do scan padrão do Nmap através do script: **nmap** <**alvo**>, onde o alvo é a máquina de IP **10.0.0.12**, o servidor principal da rede.

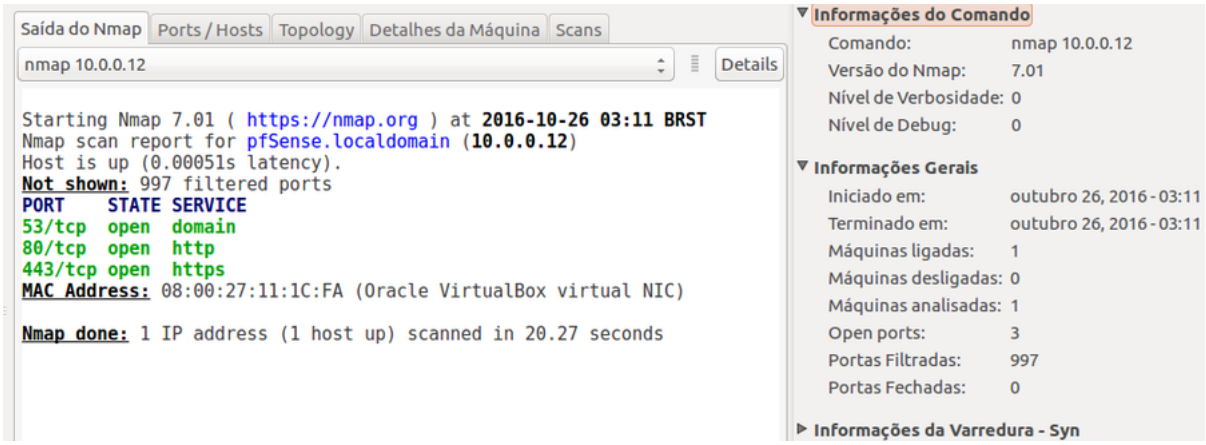
2) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de burlar a(s) regra(s) disparada inicialmente: **nmap -f** <**alvo**>

3) **Conclusão:** Finalmente, após o segundo scan, será analisado a comparação entre os resultados obtidos.

6.2.3 Ataque original

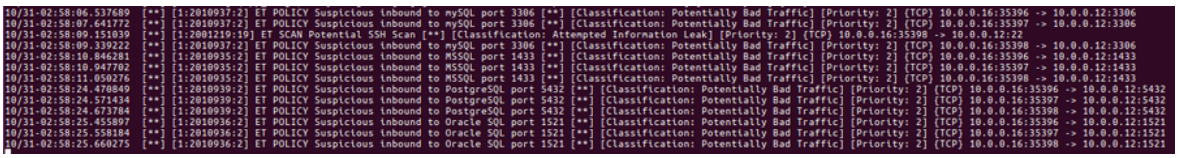
A figura 15 apresenta a chamada sem a fragmentação a partir do host atacante utilizando o Zenmap e na figura 16 o resultado (alertas) no servidor que monitora a rede com o Snort.

Figura 15 – Chamada do Zenmap sem fragmentação de pacotes



Autor

Figura 16 – Alertas resultantes do ataque sem fragmentação de pacotes



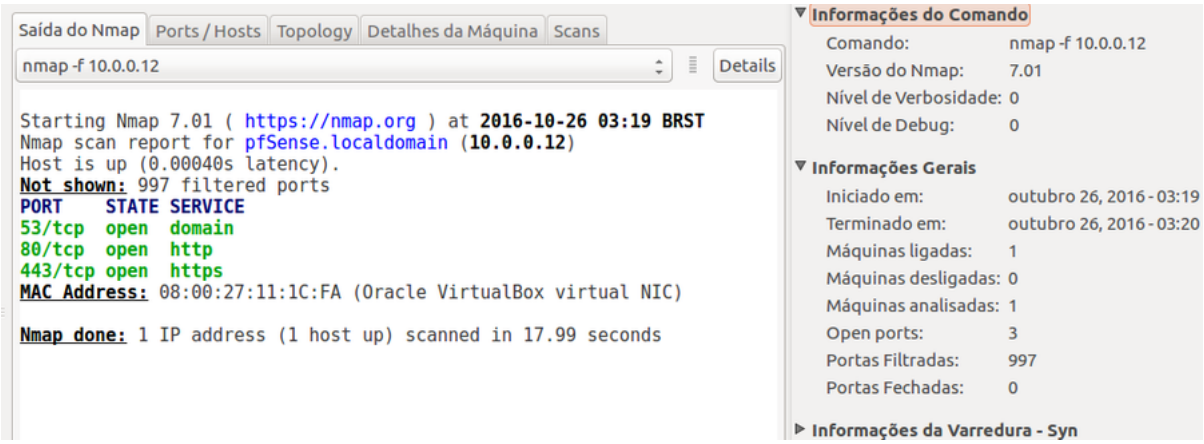
Autor

Como pode se constatar, diversos alertas foram emitidos pelo Snort, avisando que está ocorrendo um ataque contra alguma de suas máquinas protegidas.

6.2.4 Ataque adaptado

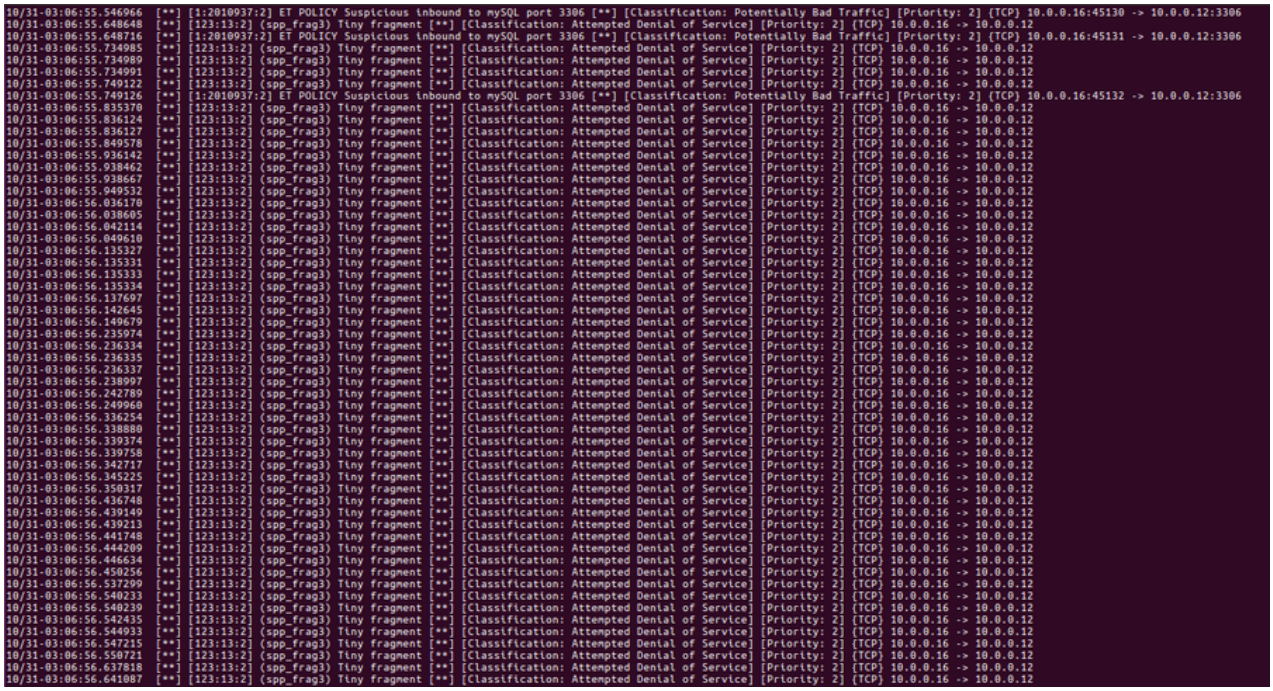
A Na figura 17 está o mesmo ataque padrão do nmap, mas agora fragmentando os pacotes enviados.

Figura 17 – Chamada do Zenmap com fragmentação de pacotes



Autor

Figura 18 – Alertas resultantes do ataque com fragmentação de pacotes



Autor

6.2.5 Conclusão

Podemos notar que somente a fragmentação não surtiu efeito na ocultação do ataque, pois ainda surtiram alertas com relação ao mapeamento de portas. E ainda teve um efeito contrário ao esperado: ao invés de minimizar a visibilidade do ataque, acabou causando uma avalanche de alertas causados pela fragmentação dos pacotes, pois o Snort interpretou esses alertas como sendo parte de uma tentativa de ataque de negação de serviço.

6.3 Evitar regras específicas

Afim de proteger ao máximo seus hosts, os IDSs utilizam regras que por vezes são genéricas, porém existem casos em que se fazem necessárias regras específicas voltadas para ataques típicos de exploradores de redes/hosts como um dos objetos deste trabalho, o Nmap.

No caso do Snort, existem diversas regras (na configuração default) que referenciam ao Nmap.

6.3.1 Experiência prática 2 - Scans do Protocolo IP

Scans do Protocolo IP permitem que você determine quais protocolos IP (TCP, ICMP, IGMP, etc.) são suportados pelas máquina-alvo.

O scan de protocolo funciona de uma forma similar a um scan UDP. Ao invés de ficar repetindo alternando o campo de número de porta de um pacote UDP, ele envia cabeçalhos de pacote IP e faz a repetição alternando o campo de protocolo IP de 8 bits. Os cabeçalhos normalmente estão vazios, e é nessa característica padrão que o Snort se baseia para detectar este tipo de scan.

6.3.1.1 Objetivo

Demonstrar o comportamento do Snort a partir do scan do protocolo IP feito pelo Nmap. Primeiramente será feito o scan padrão e em seguida será executado o mesmo exame manipulado de forma a tentar burlar a regra responsável por disparar alertas em caso de ataque.

6.3.1.2 Metodologia

Este ataque será composto por quatro etapas: Ataque original, Análise, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do scan do protocolo IP através do script: **nmap -sO** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede. Seus alertas e resultados servirão como referência posteriormente neste ataque.

2) **Análise:** Após a chamada original a partir do nmap, será disparado um alerta pelo Snort e a com ele, será possível obter a regra responsável. Em seguida, tendo a regra, vai ser possível estudar suas opções e características e articular uma forma de manipular o scan a partir dos mesmos.

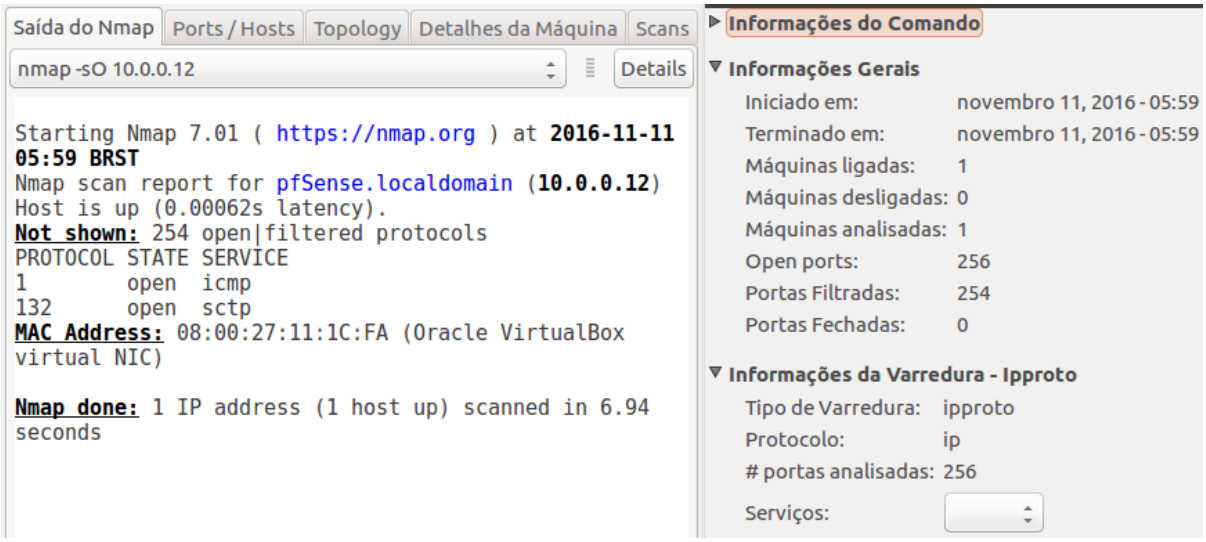
3) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de burlar a regra disparada inicialmente.

4) **Conclusão:** Finalmente, após o segundo scan, será analisada a comparação entre os resultados obtidos.

6.3.1.3 Ataque original

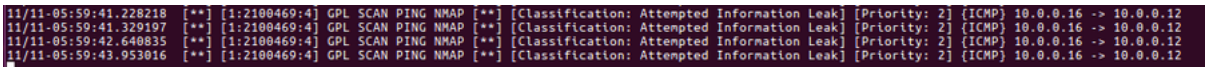
Na figura 19, é apresentada a chamada a partir do Nmap no host do atacante. Foi utilizando o scan padrão de protocolo IP, visando a máquina do servidor DHCP. Já na figura 20 está sendo exibido o comportamento do Snort ao detectar o ataque.

Figura 19 – Chamada de scan do Protocolo IP a partir do Zenmap - Antes



Autor

Figura 20 – Alertas gerados pelo scan do Protocolo IP no Snort - Antes



Autor

6.3.1.4 Análise

A partir dos alertas emitidos pelo snort pode-se perceber que todos são relacionados à mesma regra. Analisando-a isoladamente podemos notar que ela espera por um pacote com o tamanho 0 (dsize:0)

Código 6.1 – Regra do Snort para Ping do Nmap

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN
    PING NMAP"; dsize:0; itype:8; reference:arachnids,162;
    classtype:attempted-recon; sid:2100469; rev:4;)
```

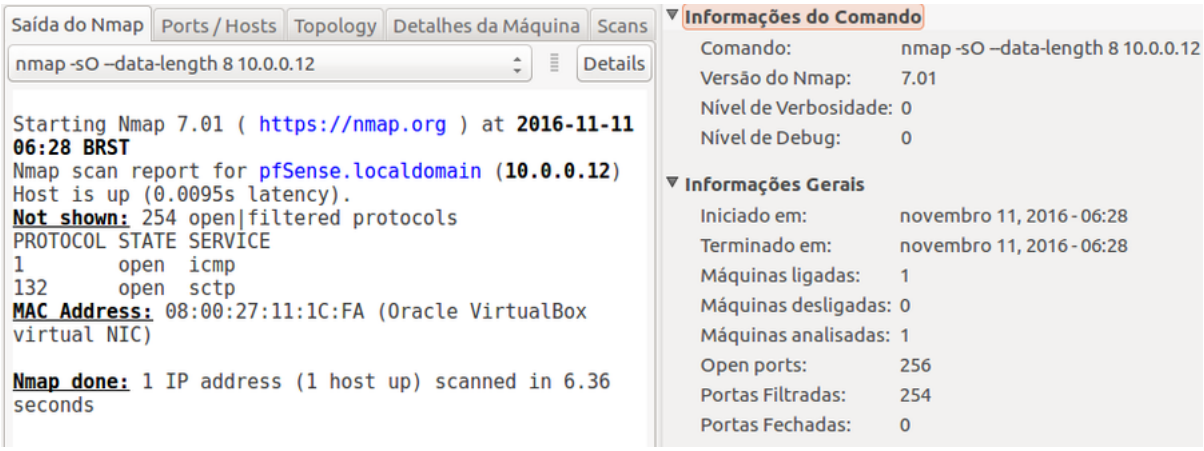
Sabendo disso, pode-se criar um script de scan como parâmetro **–data-length 8** para burlar a regra:

nmap -sO –data-length 8 <alvo>

6.3.1.5 Ataque adaptado

Na figura 21, está representado o ataque de protocolo IP adaptado, com o parâmetro definido após a análise do ataque original, visando evitar alertas no Snort. Já na figura 22, esta o log do Snort no momento do ataque adaptado.

Figura 21 – Chamada de scan do Protocolo IP a partir do Zenmap - Depois



Autor

Figura 22 – Alertas gerados pelo scan do Protocolo IP no Snort - Depois

Autor

6.3.1.6 Conclusão

Após a modificação do script do scan, incluindo o parâmetro responsável por definir o tamanho do pacote enviado e atribuindo a ele um valor, pode-se perceber que nenhum alerta foi emitido pelo Snort, além dos ruídos da rede, fazendo o ataque passar totalmente invisível aos olhos dos administradores da rede.

No lado do Snort, para prevenir que a regra seja burlada, neste caso poderia ser retirado o parâmetro que determina o tamanho do pacote como 0. Porém, teria que ser avaliada a possibilidade de aumentar falsos positivos devido a isso.

6.3.2 Experiência prática 3 - Scan nmap XMAS

O ataque “Xmas scan” tem como objetivo diferenciar os status das portas entre “closed” e “open/filtered”. Abaixo podemos observar a chamada original gerada pelo nmap, ainda utilizando pacotes TCP com os sinalizador FPU (FIN, PSH, URG) ligados Apesar de que como todas as portas já estão com status filtered/open, o ataque Xmas scan não se faria necessário neste caso.

6.3.2.1 Objetivo

Demonstrar o comportamento do Snort a partir do scan de natal (Xmas scan) feito pelo Nmap. Primeiramente será feito o scan padrão e em seguida será executado

o mesmo exame manipulado de forma a tentar burlar a regra responsável por disparar alertas em caso de ataque.

6.3.2.2 Metodologia

Este ataque será composto por quatro etapas: Ataque original, Análise, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do Scan de Natal através do script: **nmap -sX** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede. Seus alertas e resultados servirão como referência posteriormente neste ataque.

2) **Análise:** Após a chamada original a partir do nmap, será disparado um alerta pelo Snort e a partir dele, será possível obter a regra responsável. Em seguida, tendo a regra, vai ser possível estudar suas opções e características e articular uma forma de manipular o scan a partir dos mesmos.

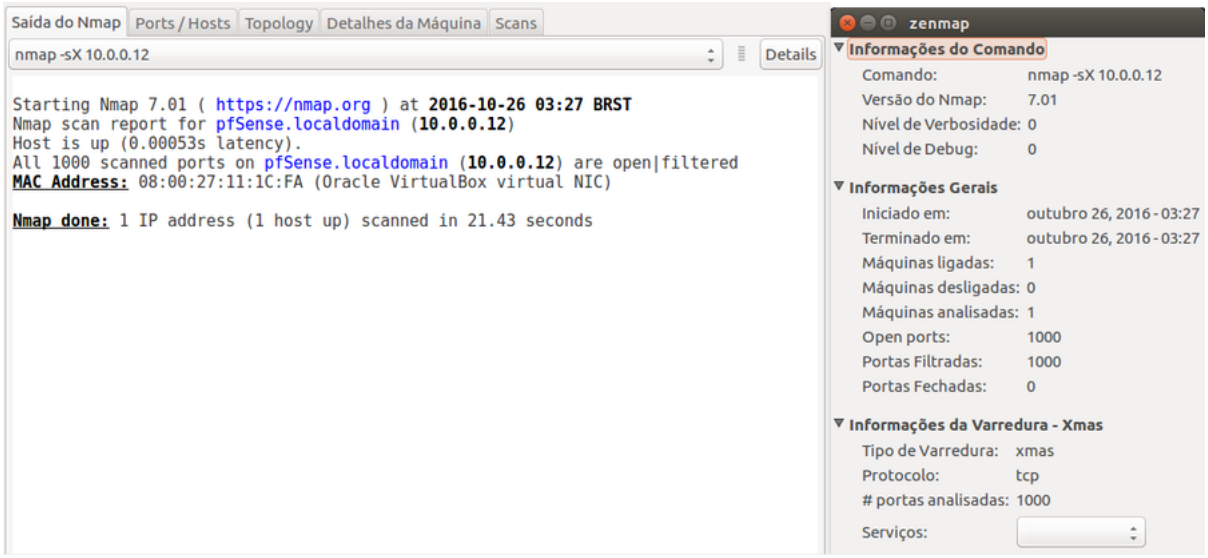
3) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de burlar a regra disparada inicialmente.

4) **Conclusão:** Finalmente, após o segundo scan, será analisada a comparação entre os resultados obtidos.

6.3.2.3 Ataque original

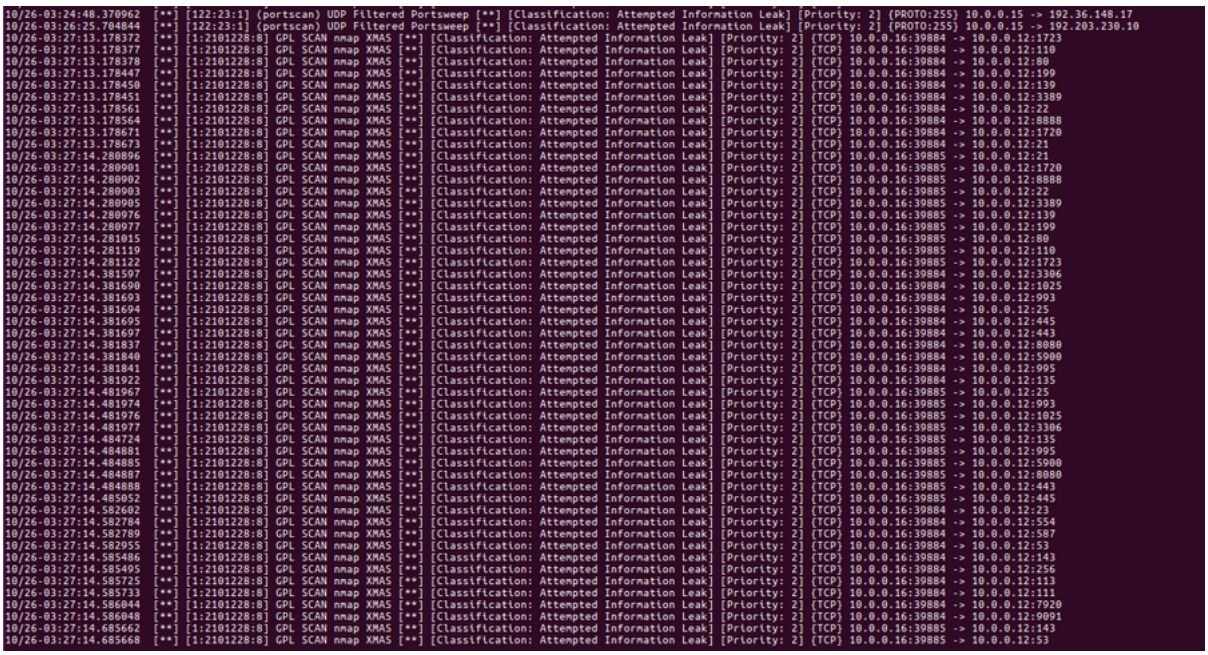
Na figura 23, é apresentada a chamada a partir do Nmap no host do atacante. Foi utilizando o scan padrão de natal (XMAS scan), visando a máquina do servidor DHCP. Já na figura 24 está sendo exibido o comportamento do Snort ao detectar o ataque.

Figura 23 – Chamada do Zenmap de XMAS scan - Original



Autor

Figura 24 – Alertas gerados no Snort pelo XMAS scan - Original



Autor

6.3.2.4 Análise

A partir do log exibido pelo Snort é possível perceber que foram disparados diversos alertas, todos referentes a mesma regra, porém apontando para scans em diferentes portas do host 10.0.0.12. Isto resume a tentativa de fazer o ataque passar despercebido em manipular o script adaptado a única regra disparada, sendo ela:

Código 6.2 – Regra XMAS scan

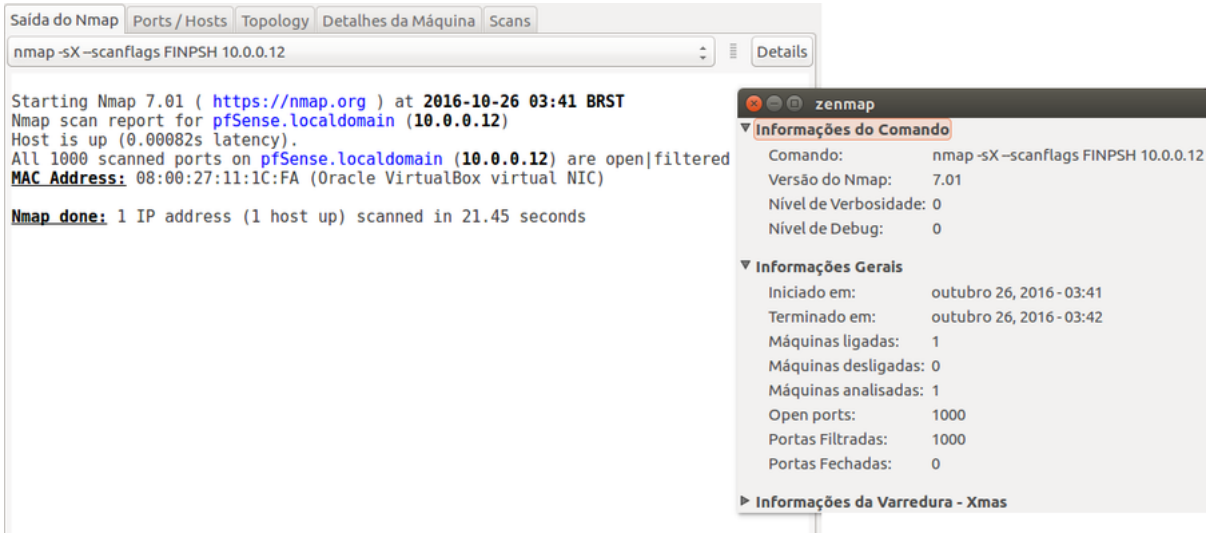
```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN
    nmap XMAS"; flow:stateless; flags:FPU,12; reference:
    arachnids,30; classtype:attempted-recon; sid:2101228; rev
    :8;)
```

Pode-se notar que uma das condições da regra é “flags:FPU,12”. Isto para o Snort significa que o pacote deve estar com os sinalizadores FIN, PSH e URG ativados, comportamento típico do ataque de natal. Sabendo disso, pode ser adicionada a opção “–scanflags FINPSH”, assim os sinalizadores FIN e PSH não serão ativados e consequentemente esta regra não será mais disparada pelo ataque.

6.3.2.5 Ataque adaptado

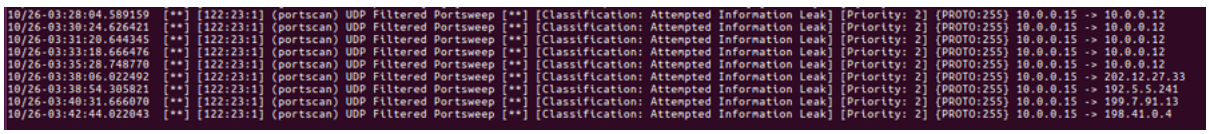
Na figura 25, está representado o ataque de scan de natal adaptado, com o parâmetro definido após a análise do ataque original, visando evitar alertas no Snort. Já na figura 26, esta o log do Snort no momento do ataque adaptado.

Figura 25 – Chamada do Zenmap de XMAS scan - Adaptado



Autor

Figura 26 – Nenhum alerta gerado no Snort pelo XMAS scan - Adaptado



Autor

6.3.2.6 Conclusão

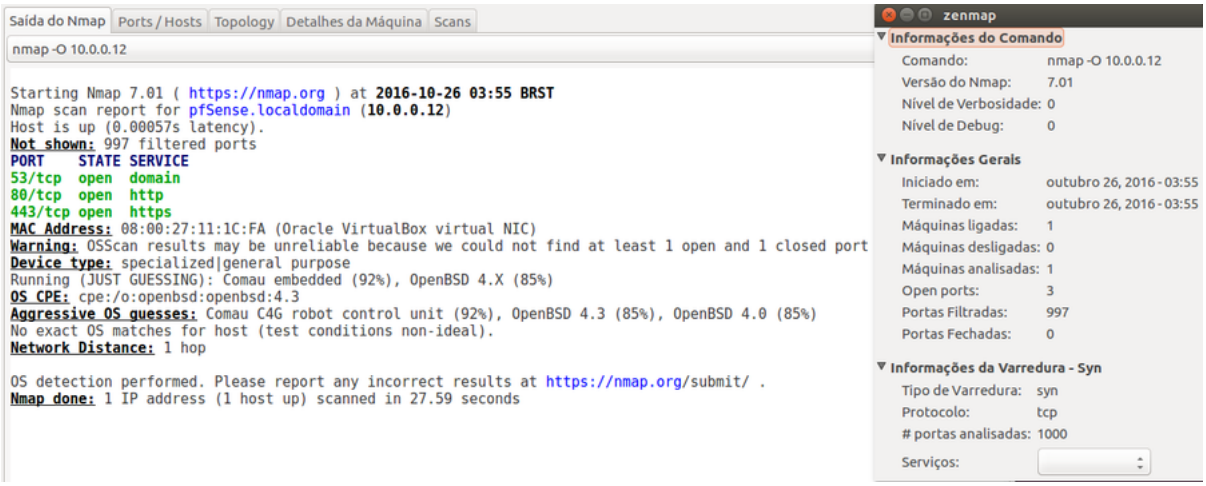
Após a modificação do script do scan, incluindo o parâmetro responsável por definir os sinalizadores que devem ser ativados no pacote e atribuindo a ele o valor para desativar os sinalizadores FIN e PSH, pode-se perceber que nenhum alerta foi emitido pelo Snort, a não ser os ruídos da rede, fazendo o ataque passar despercebido.

A solução, para o administrador da rede, seria adaptar a regra para não esperar que o pacote tenha os sinalizadores FIN e PSH ligados para emitir um alerta. Possívelmente poderia ser criada uma regra ao invés de adaptar a já existente, porém teria de ser avaliada a possibilidade de aumentar o número de falsos positivos pela mudança no critério.

6.4 Evitar funcionalidades facilmente detectáveis

Algumas funcionalidades do nmap, apesar de serem relativamente úteis para o atacante, são mais comprometedoras do que outras. A detecção de versão (-V) se conecta a muitos serviços diferentes para obter o resultado, e com isso é frequentemente motivo de alarmes nos sistemas de detecção de intrusão. Na figura 27 temos o exemplo da detecção do sistema operacional que também, é fácil de se detectar, principalmente pelo fato de utilizar pacotes e sequencias de pacotes bastante incomuns.

Figura 27 – Zenmap - Exame de Sistema operacional



Autor

Figura 28 – Snort - Exame de Sistema operacional: Diversas regras violadas

Autor

7 Distraindo um Sistema de Detecção de Intrusão

Além da abordagem da sutileza, sempre tentando agir “abaixo do radar”, existe a estratégia da distração e confusão. Este capítulo trata do uso do Nmap para falsificação de pacotes com a intenção de distração do Snort.

7.1 Experiência prática 4 - Utilizando iscas

O termo isca na área da exploração de vulnerabilidades, diferente da pescaria em que se usa uma isca para atrair a atenção do alvo, remete ao uso de vários pacotes para distrair o alvo para que o real pacote parte do ataque possa se misturar em meio a uma enxurrada de pacotes “inocentes”.

Para o uso deste trabalho, o Nmap é capaz de construir um ataque em que diversos pacotes servem como iscas inocentes vindas de vários hosts ao redor do mundo. O host alvo por sua vez não poderá pedir para os respectivos provedores por informações de cada endereço da lista pois isto de qualquer forma levaria muito tempo sendo que somente um dentre todos é o real atacante.

No Nmap a opção responsável por produzir este comportamento é a opção -D seguida da lista de hospedeiros separados por vírgula. Existe ainda o atributo ME que determina em qual posição o verdadeiro hospedeiro deve aparecer. Isto é importante pois a maioria dos detectores de exames de portas costumam reportar somente as primeiras 5 fontes de exames, para evitar overflow de seus registros com pacotes iscas.

A opção RND serve para gerar IPs aleatórios, usado no ping e também no exame de portas.

7.1.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente será feito o scan padrão e em seguida será executado o mesmo exame manipulado de forma a tentar distrair o Snort e disparar alertas falso-positivos.

7.1.2 Metodologia

Este ataque será composto por três etapas: Ataque original, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP

10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão.

2) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -D RND:5** <**alvo**>

3) **Conclusão:** Finalmente, após o segundo scan, será analisada a comparação entre os resultados obtidos.

7.1.3 Ataque original

Na figura 29, é apresentada a chamada a partir do Nmap no host do atacante. Foi utilizando o scan padrão do Nmap, visando a máquina do servidor DHCP. Já na figura 30 está sendo exibido o comportamento do Snort ao detectar o ataque.

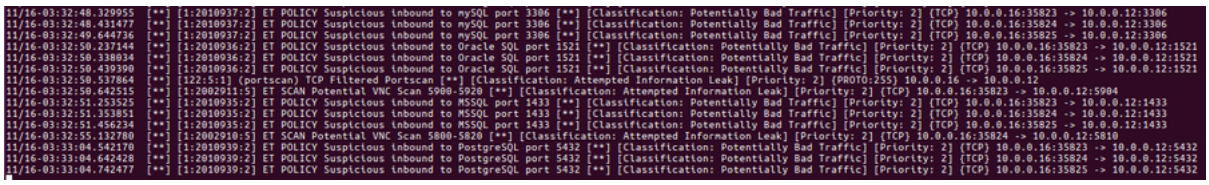
Figura 29 – Zenmap - Ataque padrão sem iscas

```
nmap -v -Pn 10.0.0.12
Starting Nmap 7.01 ( https://nmap.org ) at 2016-11-13 08:21 BRST
Initiating ARP Ping Scan at 08:21
Scanning 10.0.0.12 [1 port]
Completed ARP Ping Scan at 08:21, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:21
Completed Parallel DNS resolution of 1 host. at 08:21, 0.00s elapsed
Initiating SYN Stealth Scan at 08:21
Scanning pfSense.localdomain (10.0.0.12) [1000 ports]
Discovered open port 53/tcp on 10.0.0.12
Discovered open port 443/tcp on 10.0.0.12
Discovered open port 80/tcp on 10.0.0.12
Completed SYN Stealth Scan at 08:21, 18.11s elapsed (1000 total ports)
Nmap scan report for pfSense.localdomain (10.0.0.12)
Host is up (0.00044s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:11:1C:FA (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 18.36 seconds
           Raw packets sent: 3009 (132.380KB) | Rcvd: 17 (732B)
```

Autor

Figura 30 – Snort - Alertas de um ataque padrão sem iscas



Autor

7.1.4 Ataque adaptado

Na figura 31, está representado o ataque padrão do Nmap, adaptado, com o parâmetro de distração através de iscas, visando esconder o ataque através de alertas falsos no Snort. Já na figura 32, esta o log do Snort no momento do ataque adaptado.

Figura 31 – Zenmap - Ataque utilizando iscas de cinco IPs aleatórios

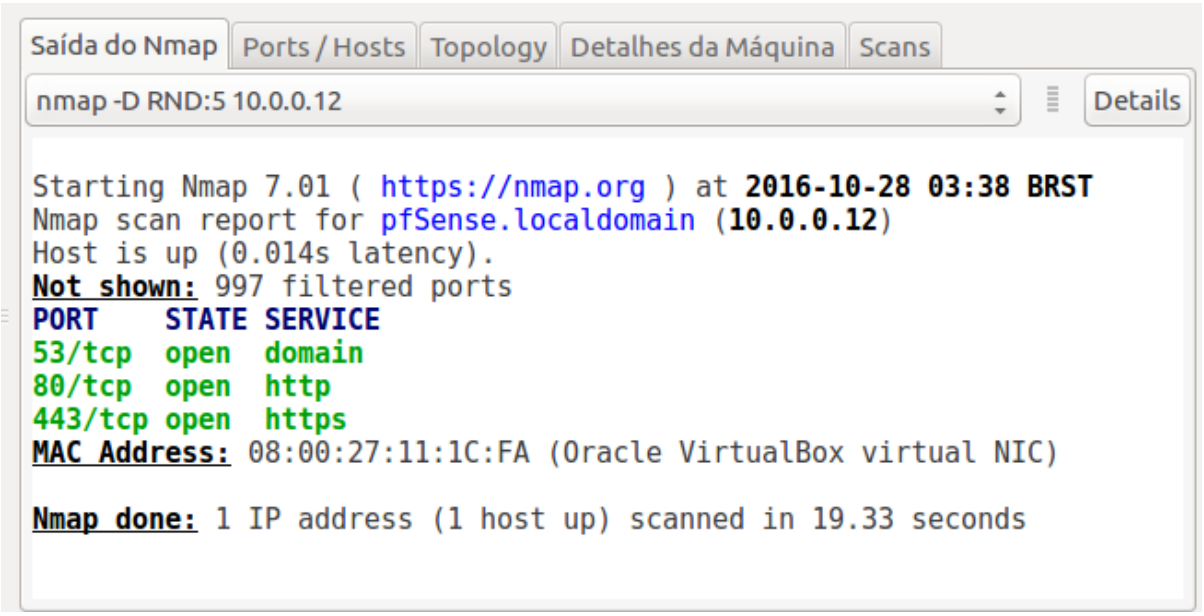
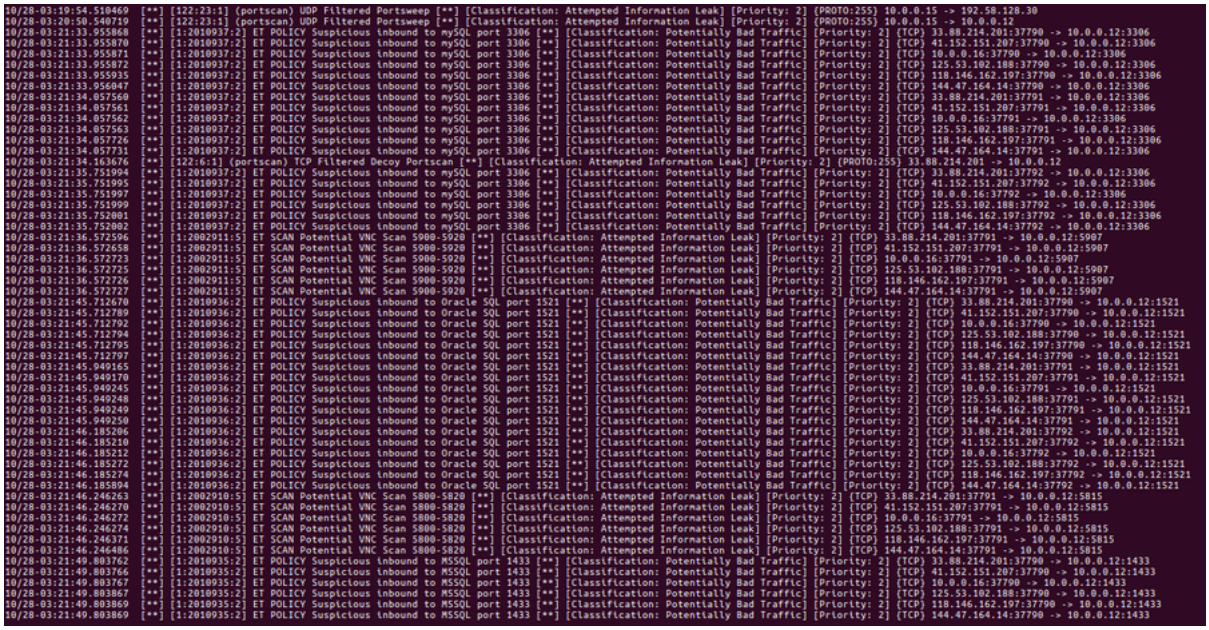


Figura 32 – Snort - Alertas do ataque utilizando iscas de cinco IPs aleatórios



Autor

7.1.5 Conclusão

Após a comparação entre os dois resultados, antes e depois da modificação do script, foi possível observar que o comportamento esperado foi atingido, de forma que os alertas gerados pelo ataque real, proveniente do IP 10.0.0.16, foram cercados por falsos alertas que apontam para IPs que não são do atacante. Isto, em uma situação real, causaria um alerta para os administradores, porém seria bastante difícil a busca pelo real responsável.

7.2 Experiência prática 5 - Simulação de exames de porta

Apesar de utilizar iscas para inundar o alvo com pacotes cumpra com o seu papel de esconder a verdadeira origem do ataque, ficará óbvio para os administradores da rede alvo que a mesma está sendo alvo de ataques pelo uso de iscas. Existe outra estratégia que pode ser utilizada em conjunto com outros para distração e confusão dos responsáveis pela rede alvo: a simulação de exames de porta.

Ela é basicamente o exame de portas simulando ser feito por um outro host que não o do real atacante. Desta forma, o alerta (que provavelmente será gerado) do IDS apontará para um host que foi determinado pelo atacante. Porém, da mesma forma que ele indica um ataque partindo do host simulado, ele retorna as possíveis informações relevantes para o mesmo host que não é o de origem real do atacante, portanto neste tipo de distração não existe a vantagem de conseguir informações sensíveis e úteis sobre o alvo.

7.2.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente será feito o scan padrão e em seguida, será executado o mesmo exame manipulado, de forma a tentar distrair o Snort e fazê-lo disparar alertas apontando para uma máquina que não a do real atacante.

7.2.2 Metodologia

Este ataque será composto por três etapas: Ataque original, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão.

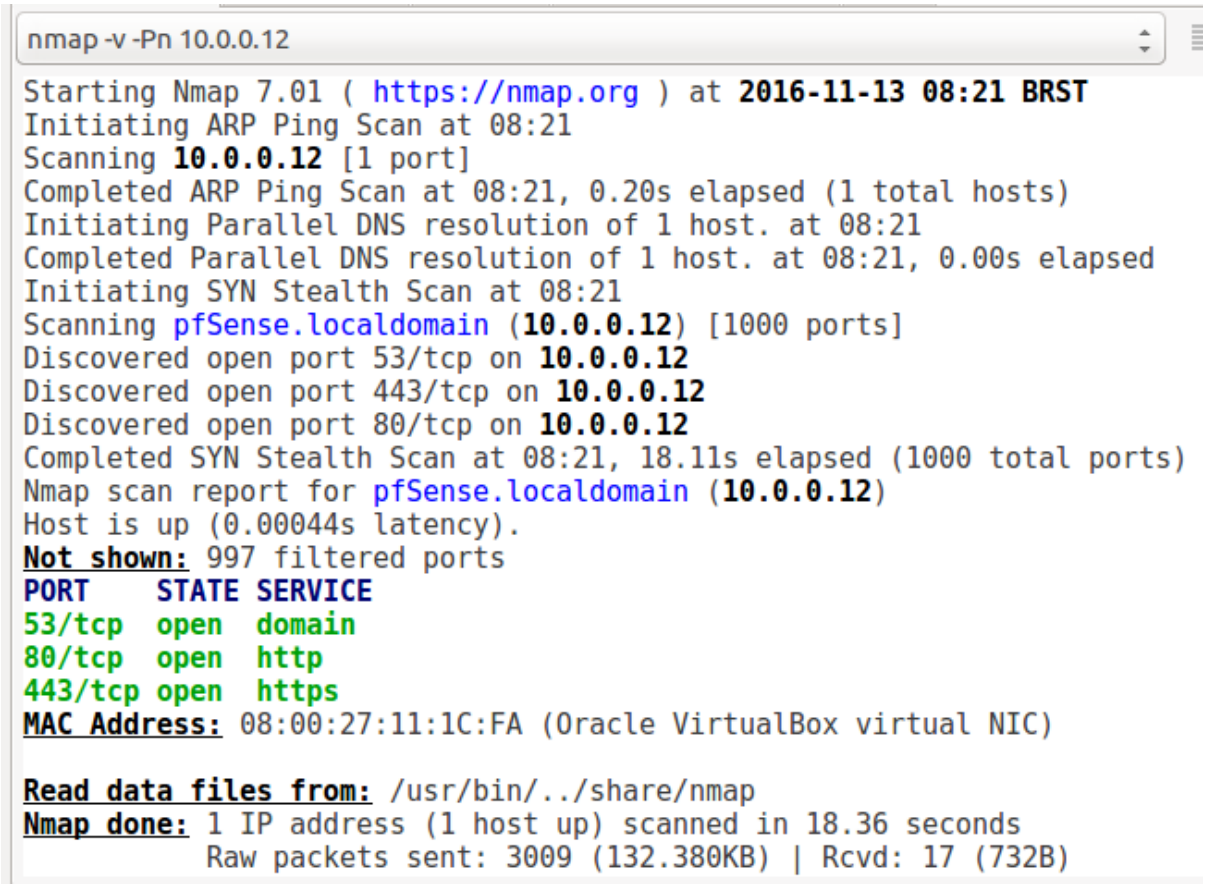
2) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -Pn -e enp0s8 -S** <**falsa origem**> <**alvo**>, onde o parâmetro -e é a interface a ser usada e a falsa origem é o IP aleatório: **216.20.149.10**

3) **Conclusão:** Finalmente, após o segundo scan, será analisada a comparação entre os resultados obtidos.

7.2.3 Ataque original

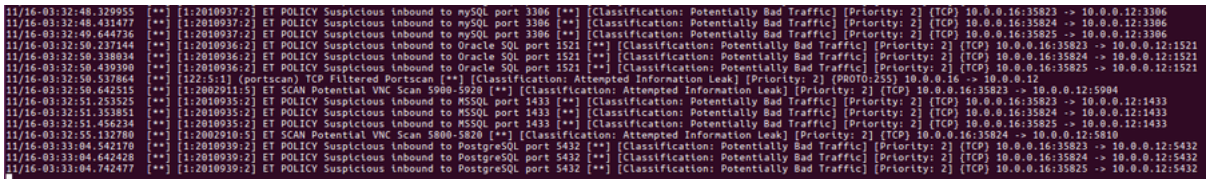
Na figura 33, é apresentada a chamada a partir do Nmap no host do atacante. Foi utilizando o scan padrão de protocolo IP, visando a máquina do servidor DHCP. Já na figura 34 está sendo exibido o comportamento do Snort ao detectar o ataque.

Figura 33 – Zenmap - Ataque original partindo do atacante 10.0.0.16



Autor

Figura 34 – Snort - Alertas do ataque original partindo do atacante 10.0.0.16

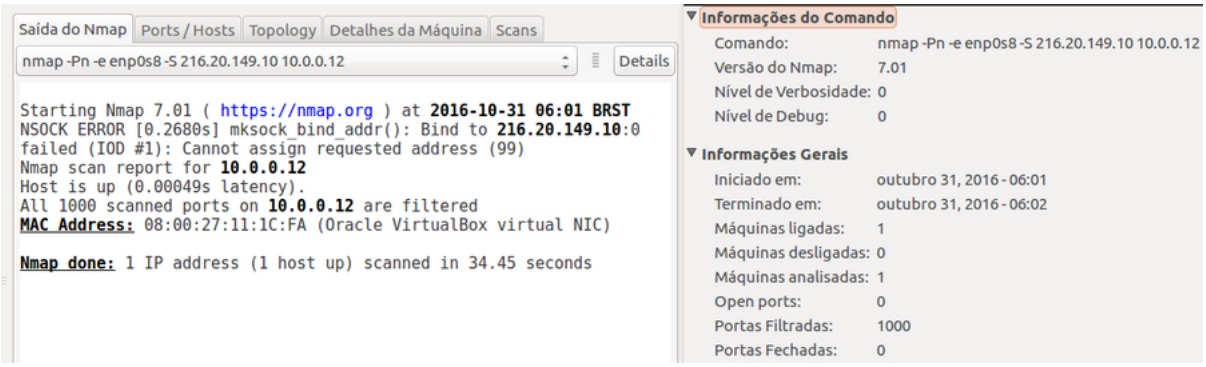


Autor

7.2.4 Ataque adaptado

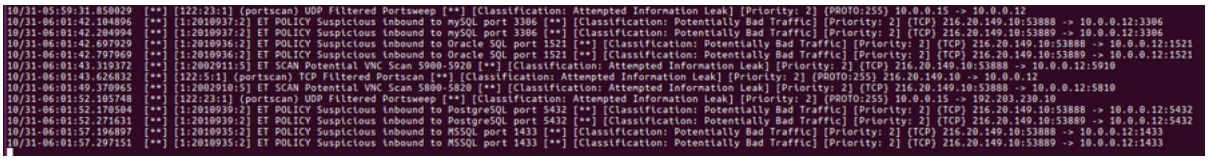
Na figura 35 está representado o ataque padrão do Nmap, adaptado, com o parâmetro de distração através simulação de exame de portas, visando distrair os administradores da rede, através de alertas falsos no Snort. Já na figura 36, esta o log do Snort no momento do ataque adaptado.

Figura 35 – Zenmap - Ataque simulado a partir do IP 216.20.149.10



Autor

Figura 36 – Snort - Ataque simulado a partir do IP 216.20.149.10



Autor

7.2.5 Conclusão

Após observar ambos os ataques, antes e depois da manipulação através do parâmetro -S, foi possível comprovar que ao fabricar pacotes endereçados por um IP aleatório, o Snort gera alertas falsos já que apontam para uma máquina que não é a do atacante. Neste caso, os alertas são falsos pois, diferente do ataque ocioso, não são aproveitadas informações por parte do atacante, o único objetivo é distrair.

7.3 Experiência prática 6 - Exame ocioso

Em 1998 um pesquisador de segurança chamado Antierez desenvolveu uma técnica que permitiria exames de portas de forma não fosse possível para o alvo descobrir diretamente de onde estariam vindo os verdadeiros ataques, ao mesmo tempo que obteria resultados precisos sobre as portas de TCP. Esta técnica ficou conhecida como exame ocioso, pois o atacante forja sua identidade de forma que pareça que uma máquina zumbi inocente fez o exame.

Este tipo de exame se baseia em três princípios básicos do comportamento de máquinas em redes:

- Ao enviar um pacote SYN (estabelecimento de sessão) a uma determinada porta a máquina responsável responderá com um pacote SYN/ACK (requisição de

sessão reconhecida) caso a porta esteja aberta ou RST (resetar) se a porta estiver fechada. Com isso é possível determinar se uma porta TCP está aberta.

- Ao receber um pacote SYN/ACK que não foi solicitado por ela responderá com outro pacote RST. Caso receba um RST não solicitado ele simplesmente será ignorado.
- Todo o pacote IP tem embutido um número de identificação. Como este número na grande maioria das vezes é sequencial e incrementado de acordo com o número de pacotes que a máquina envia, é possível determinar quantos pacotes foram enviados desde a última prova feita pelo atacante.

7.3.1 Passo a passo

Fundamentalmente, após conseguir uma máquina zumbi, o exame ocioso é feito por três passos repetidos para cada porta alvo, sendo eles:

1) Determinar o ID de IP da máquina zumbi;
2) Forjar um pacote SYN a partir da máquina zumbi e enviá-lo ao alvo na determinada porta. A reação do alvo a partir do recebimento do pacote fará com que o ID de IP do zumbi seja incrementado dependendo do estado da porta alvo;
3) O atacante deve provar novamente o ID de IP do zumbi. O estado da porta será determinado a partir da comparação do ID de IP obtido no passo 1.

Ao final do terceiro passo, o ID de IP da máquina zumbi terá sido incrementado em um ou dois.

No caso de ter sido incrementado em um significa que o zumbi não enviou nenhum pacote a não ser a própria resposta à prova inicial do atacante (passo 1) e isso determina que a porta não está aberta.

Já se o ID de IP tenha incrementado em dois significa que o zumbi enviou um pacote entre as duas provas feitas pelo atacante (passos 1 e 3). Este pacote extra significa que a máquina alvo enviou um pacote SYN/ACK ao zumbi em resposta ao pacote SYN forjado pelo atacante (passo 2) e portanto prova que a porta alvo está aberta.

Em casos de incremento maiores do que dois podem significar que a máquina zumbi não possui incremento sequencial de ID de IP ou então que ela pode já estar em comunicação não relacionada ao exame ocioso.

7.3.2 Executando um exame ocioso

7.3.2.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente será feito o scan padrão e em seguida será executado o mesmo exame manipulado de forma a tentar distrair o Snort e disparar alertas apontando para uma máquina que não a do real atacante.

7.3.2.2 Metodologia

Este ataque será composto por três etapas: Ataque original, Ataque adaptado e Conclusão.

1) **Ataque original:** Inicialmente será executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão;

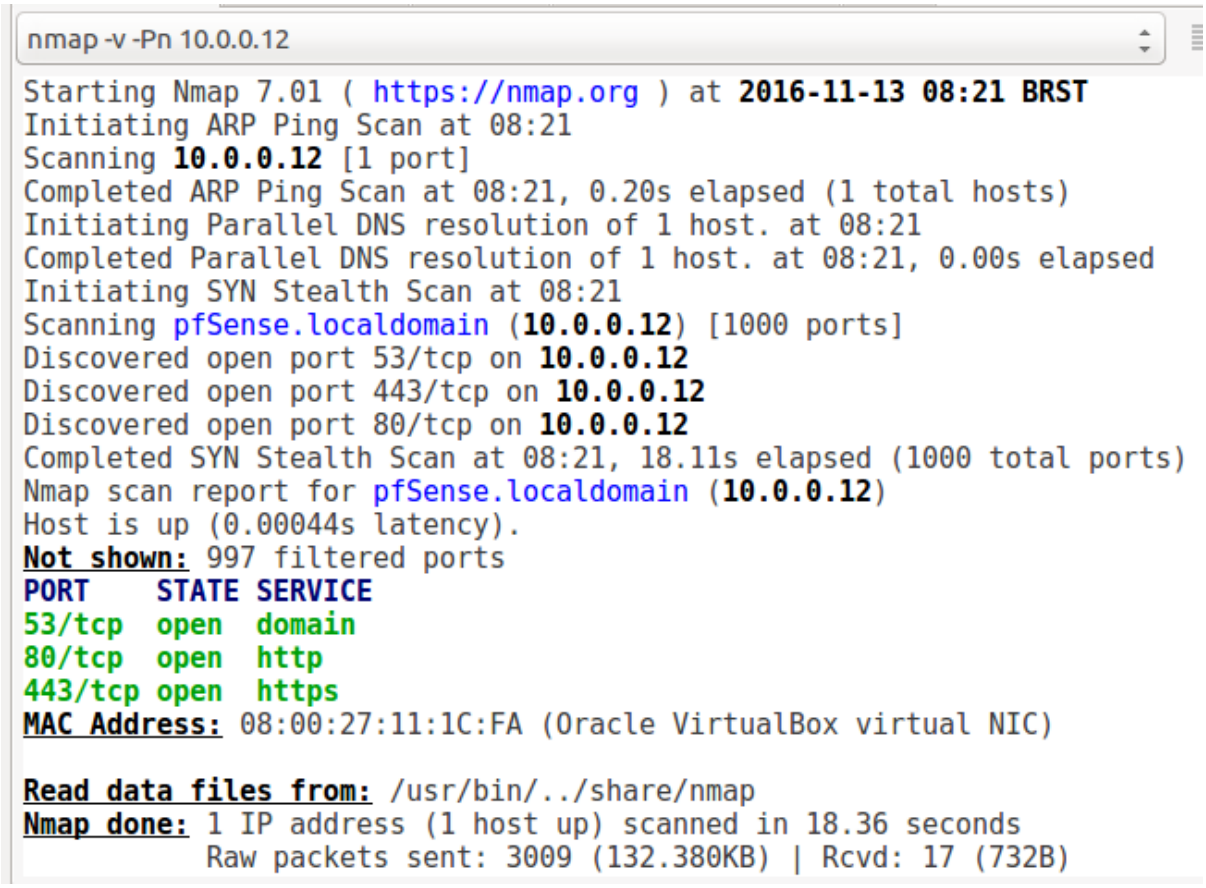
2) **Ataque adaptado:** Com a estratégia de manipulação, será feito um segundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -sI** <**máquina zumbi**>**-p 80** <**alvo**>, onde a máquina zumbi é o IP: **10.0.0.18** e o alvo continua sendo **10.0.012;**

3) **Conclusão:** Finalmente, após o segundo scan, será analisada a comparação entre os resultados obtidos.

7.3.2.3 Ataque original

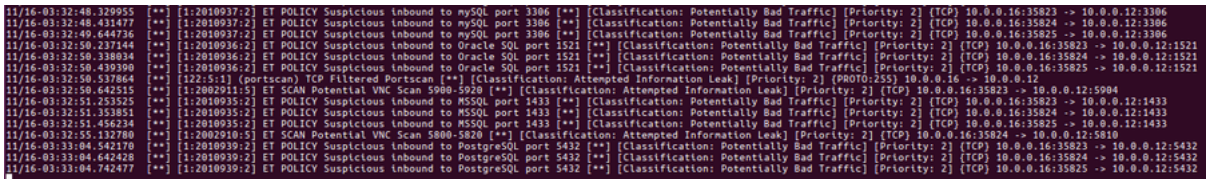
Na figura 37, é apresentada a chamada a partir do Nmap no host do atacante. Foi utilizando o scan padrão do Nmap, visando a máquina do servidor DHCP. Já na figura 38 está sendo exibido o comportamento do Snort ao detectar o ataque.

Figura 37 – Zenmap - Ataque padrão



Autor

Figura 38 – Snort - Alertas de um ataque padrão

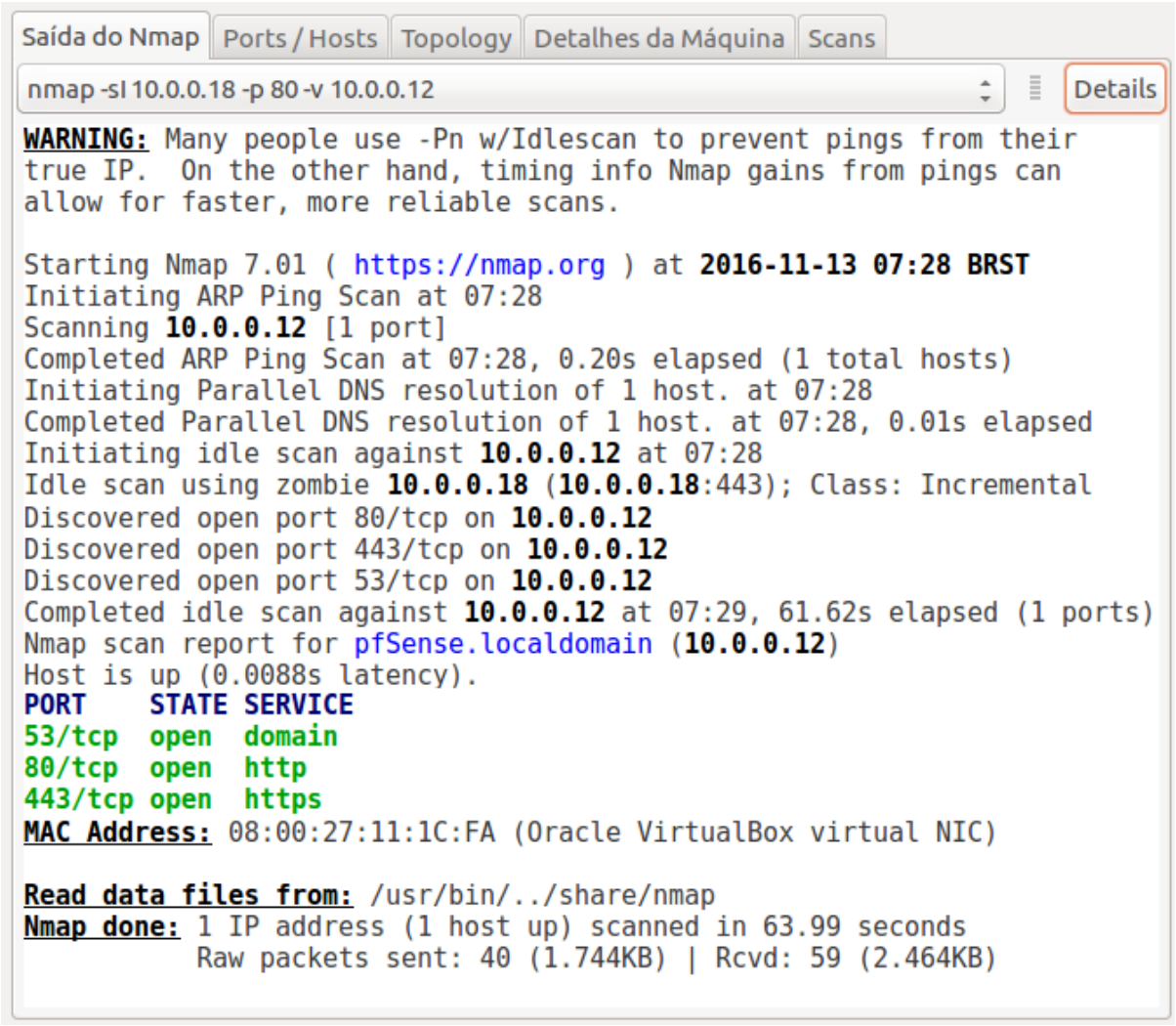


Autor

7.3.2.4 Ataque adaptado

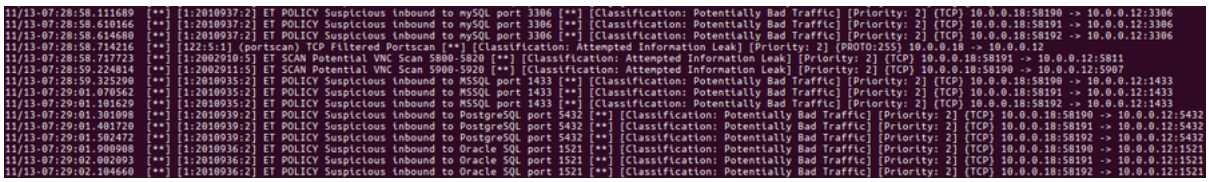
Na figura 39, está representado o ataque original do Nmap já com o parâmetro do ataque ocioso. Para isso está sendo utilizando uma terceira máquina, como ferramenta intermediária, para assim obter as informações, sem que sejam disparados alertas direcionados ao atacante real, mas sim para a terceira máquina. Já na figura 40, esta o log do Snort no momento do ataque adaptado.

Figura 39 – Zenmap - Exame ocioso



Autor

Figura 40 – Snort - Alertas gerados pelo exame ocioso



Autor

7.3.2.5 Conclusão

O objetivo do ataque ocioso, é ser a junção entre o uso de iscas e o de simulação de portas, pois é utilizado um IP diferente do atacante, porém ainda assim ele consegue as informações do alvo. Analisando os dados de antes e depois dos ataque, comprova-

se que o objetivo é atingido, pois as informações são as mesmas e além disso a identidade do atacante ficou oculta, mesmo que seu ataque tenha gerado alertas.

7.4 Representação de DNS

Dependendo do nível de critério dos administradores de uma determinada rede ou se os IDSs da mesma tiverem uma sensibilidade elevada, pode ser que até ataques preparados para serem invisíveis sejam detectados devido a resolução de DNS feita pelo Nmap. Isto pode acontecer pois mesmo em um exame não tão intrusivo como a listagem de portas o Nmap faz a resolução de DNS reverso por omissão para cada hospedeiro responsivo. Estas provas indicarão o DNS configurado para a máquina onde está o Nmap, dependendo dos casos pode ser uma máquina separada, mantida pelo provedor de acesso porém, às vezes, pode ser o seu próprio sistema.

Existem duas abordagens tradicionais para evitar este problema:

1) Utilizar flag (-n) para suprimir a resolução do DNS reverso por parte do Nmap;
2) Utilizar servidores de DNS abertos para consulta para obter de lá as informações sensíveis do alvo (–dns-servers <servidor DNS>).

Na primeira opção, o ataque vai passar invisível mesmo para o contexto de alta sensibilidade citado anteriormente mas ao mesmo tempo vem com a falta das informações do DNS reverso.

Já a segunda opção se utiliza de um mecanismo alternativo e muito mais sofisticado para contornar a situação. Sabendo-se que muitos dos servidores DNS estão abertos para consultas recursivas, é possível especificar um ou mais destes servidores para que sejam usados nas eventuais resoluções de DNS reverso.

8 Conclusão e trabalhos futuros

Neste trabalho, foram apresentadas algumas das estratégias mais comumente utilizadas por atacantes afim de comprometer uma rede monitorada por um IDS. Apesar de não ser possível cobrir todos os casos, estes exames expostos podem servir como início de uma auditoria de segurança de uma rede.

A partir destes dados, e principalmente dos experimentos práticos, foi possível perceber que, a maior parte das tentativas de obtenção de dados sensíveis, se utiliza de limiares ou brechas presentes na rede devido a omissão durante a configuração. Portanto, executando os mesmos ataques feitos neste estudo, em uma rede real, será possível encontrar as principais brechas e adaptar as regras de segurança do IDS para captar e alertas estes exames, como no caso do exame de natal adaptado, sem os sinalizadores, seria possível adaptar a regra para não exigir que os sinalizadores estejam ligados para disparar alertas.

Porém, também é necessário ter em mente que, mesmo os ataques que são notificados, nem sempre são ataques reais ou apontam para o atacante real, como o Exame ocioso. Para estes casos, o ideal é criar um monitoramento baseado em número de alertas, pois assim, os administradores não vão gastar mais tempo do que o necessário analisando dados, e mesmo assim manteriam a segurança na rede.

Finalmente, ficou evidente que, o IDS deve ser utilizado em conjunto com outras ferramentas, pois apesar de ser efetivo no monitoramento, pode ser otimizado no aspecto de filtragem de falsos alertas e ruídos de rede. Mais especificamente, é imprescindível que ó Snort seja utilizado associado com algum firewall, principalmente para controlar o fluxo de portas que não podem ser fechadas devido ao seu uso.

Com relação as limitações, foi necessário utilizar uma rede virtual baseada em máquinas virtuais, devido a não disponibilidade de uma estrutura física em um laboratório com tráfego real. Além disso, ficou faltando o uso de uma interface gráfica para uma melhor interpretação dos alertas emitidos pelo Snort. Isto foi causado pela falta de opções atuais responsáveis por esta parte do monitoramento, já que os mais famosos existentes, Snorby e Sguil estão considerados ultrapassados e não tem mais suporte.

8.1 Trabalhos futuros

Os trabalhos futuros que podem ser citados são:

- Utilizar os ataques e experiências práticas para definir adaptações para as regras

do Snort em uma rede real.

- Estender o estudo para outros sistemas de detecção de intrusão como o Suricata e o RealSecure.
- Adaptar o estudo feito por este trabalho para o Snort em forma de sistema de prevenção de intrusão (IPS).
- Incluir uma interface gráfica para uma melhor interpretação dos resultados e alertas emitidos pelo Snort.

Referências

DENNING, D. E. An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, Oakland, CA, USA, v. 13, n. 02, p. 222 – 232, Abril 1986. Citado na página 21.

LYON, G. *NMap Usage*. Disponível em: <https://svn.nmap.org/nmap/docs/nmap.usage.txt>. Acesso em: 27/10/2016. Citado na página 34.

LYON, G. *Exame de redes com Nmap*. [S.l.]: Ciência Moderna, 2009. Citado 2 vezes nas páginas 16 e 30.

M., C. J. *Analisador comportamental de rede*. 2006. Dissertação (Mestrado) — a Faculdade de Ciências de Lisboa. Citado na página 23.

MELO, S. *Anatomia de um teste de penetração*. [S.l.]: Editora Alta Books, 2008. Citado 2 vezes nas páginas 8 e 47.

NAKAMURA, E. T. *Segurança de Redes em Ambientes Cooperativos*. [S.l.]: Novatec, 2007. Citado na página 21.

NUNES, L. F. *Auditoria com Testes de Penetração para Ambientes Seguros*. 2014. Monografia (Sistemas de Informação) — UNIVERSIDADE FEDERAL DE SANTA CATARINA. Citado na página 17.

REHMAN, U. R. *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID*. 2013. Disponível em: <http://ptgmedia.pearsoncmg.com/images/0131407333/downloads/0131407333.pdf>. Citado na página 22.

SILVA, F. B. da. *Auditoria com Testes de Penetração para Ambientes Seguros II*. 2015. Monografia (Sistemas de Informação) — UNIVERSIDADE FEDERAL DE SANTA CATARINA. Citado na página 17.

Apêndices

APÊNDICE A – Artigo

Subvertendo um sistema de detecção de intrusão: Caso prático utilizando Snort e Nmap

Mauricio Branco Biazus 1, João Bosco Mangueira Sobral 1

1 Instituto de Estatística e Informática – Universidade Federal de Santa Catarina
Campus Reitor João David Ferreira Lima, s/n - Trindade, Florianópolis - SC, 88040-900

mauriciobranco@live.com, jbmsobral@gmail.com

Abstract. *The Intrusion Detection System (IDS) is a security layer that, from the analysis and monitoring of events, on the network or its host, generates alerts from pre-established rules in order to warn those res- ponsible and facilitate the taking And countermeasures. One of the most used IDS by the community is Snort, which has its code open and is based on rules many of them set by the community itself.*

In the counterflow, are vulnerability exploiters and network mappers. They are usually based on sending packets in order to get information relevant to attacks. The best-known network mapper is Nmap. Nmap, like Snort, is open source maintained by a relatively large community.

Even though the two tools are widely known, their thresholds bring, for those who use them, loopholes according to the established settings.

Resumo. *O Sistema de detecção de Intrusão (IDS) é uma camada de segurança que, a partir da análise e monitoramento de eventos, na rede ou em seu hospedeiro, gera alertas a partir de regras pré- estabelecidas afim de avisar os responsáveis e facilitar a tomada de decisão e contramedidas. Um dos mais utilizados IDS pela comunidade é o Snort, que tem seu código aberto e é baseado em regras muitas delas estabelecidas pela própria comunidade.*

No contra fluxo, estão os exploradores de vulnerabilidades e mapeadores de redes. Normalmente se baseiam no envio de pacotes afim de conse- guir informações relevantes para ataques. O mais conhecido mapeador de rede é o Nmap. O Nmap, como o Snort, é de código aberto mantido por uma relativamente grande comunidade.

Mesmo as duas ferramentas sendo amplamente conhecidas, seus limia- res trazem, para quem as usa, brechas de acordo com as configurações estabelecidas.

A.1 Introdução

A segurança na Internet sempre foi motivo de preocupação para todos os usuários, tanto para usuários domésticos, quanto para usuários corporativos e com o aumento exponencial das redes e do seu uso, as atenções estão cada vez mais voltadas para os campos de prevenção, contenção e contra-ataque as ameaças e usuários maliciosos. A lista de ameaças inclui vírus de computa- dor, worms, spywares, exploits e os próprios Crackers. Esta lista só não é maior do que a de ferramentas e técnicas utilizadas para tentar conter essa avalanche de ameaças: Firewalls, antivirus, anti-spywares, anti-rootkits, verificadores de integridade, configuração segura de servidores, entre muitos outros. Nesta lista de armas para manter o nível de segurança aceitável estão os Sistemas de Detecção de Intrusão, que têm como princípio básico procurar antecipar possíveis tentativas de acesso malicioso através de monitoramento e detecção de padrões na rede de ataques conhecidos ou de mudanças no padrão de comportamento do uso da rede.

A análise do tráfego de uma rede é fundamental para manutenção de um bom funcionamento da rede, tanto em questão de segurança como também em questão de qualidade, já que ataques também podem comprometer direta ou indiretamente o desempenho do ambiente onde está ocorrendo.

A.2 Sistemas de Detecção de Intrusão

A utilização de métodos de detecção de intrusão permite recolher informa- ção sobre tipos de ataques já conhecidos e identificar tentativas de ataque à rede ou a algum servidor em particular. A informação recolhida servirá essencialmente para a tomada de decisões que levem à proteção do alvo do ataque e também poderá constituir uma base informativa para uma ação legal.

Um sistema global de segurança, consiste num conjunto de ferramentas que incluem:

- Firewalls
- Sistemas de Detecção de Intrusão (IDS)

- Sistemas de Avaliação de Vulnerabilidade

Estas ferramentas deverão trabalhar em conjunto e partilhar informação de uma forma dinâmica. De acordo com [Nakamura, 2007], um sistema de detecção de intrusão funciona de acordo com uma série de funções que, trabalhando de modo inte- grado, são capazes de detectar, analisar e responder as atividades suspeitas.

Os Sistemas de Detecção de intrusão em Redes de computadores (NIDS, Network Intrusion Detection System) são utilizados para monitoramento do tráfego de dados de uma rede ou de um segmento de rede. A análise é realizada em dados coletados da rede, ou em base de dados disponíveis ao NIDS.

A.2.1 Snort

O Snort é uma ferramenta de código aberto de detecção de intrusão de rede (NIDS) disponível gratuitamente. É essencialmente um IDS baseado em regras (conjunto de assinaturas), no entanto existem plug-ins para detectarem anomalias nos cabeçalhos de protocolo. As regras são armazenadas em ficheiros e podem ser modificados por um editor de texto simples. Os ficheiros de regras são referenciados no ficheiro de configuração snort.conf. No momento em que a aplicação inicia, são criadas as respectivas estruturas de dados internas, que irão aplicar as regras aos dados capturados.

O Snort pode ser configurado em três modos:

- Modo Sniffer: apenas lê os pacotes da rede mostrando-os de forma contí- nua no console.
- Modo packet logger: é semelhante ao anterior, com a diferença de redireci- onar o output para disco.
- Modo network intrusion detection: permite a análise do tráfego da rede tentando encontrar algum padrão descrito nas regras previamente estabe- lecidas e atuar em conformidade (alertas).

O Snort já vem com um conjunto vasto de regras para detectar atividades de intrusão. A esse conjunto podem-se adicionar regras próprias ou remover algu- mas das pré-definidas. Adicionalmente às regras fornecidas pelo Snort, poderão ser implementadas outras regras para fazer face a requisitos específicos de um determinado ambiente.

A.2.1.1 Regras do Snort

Uma das melhores características do Snort é o seu módulo de regras e as regras em sí. O módulo de regras disponibiliza uma linguagem relativamente extensa, e permite com que os próprios usuários possam escrever as suas próprias regras, afim de que estas se adaptem ao ambiente em que o Snort está alocado.

Uma regra pode ser quebrada em duas partes principais: O cabeçalho e as opções. O cabeçalho contém a ação que deve ser tomada, o protocolo em que a regra se aplica e os endereços e portas de destino e de origem.

A.3 Nmap

O Nmap é um utilitário livre e de código aberto que, tendo o apoio de seu desenvolvimento na comunidade, cresceu até se tornar o scanner de segurança de redes mais popular do mundonotaCom base na frequência de download, número de alcances no Google e na parada de “popularidade” de software do Freshmmeal.net. (Dados de 2009). Com sua exploração baseada em pacotes de IP, o Nmap consegue:

- Determinar quais hospedeiros estão disponíveis.
- Detectar SO - Remotamente determina o sistema operacional e caracterís- ticas do hardware do alvo.
- Executar exame ocioso por ID de IP.
- Interação com scripts com o alvo.
- Examinar portas - Detecção da versão/serviço utilizando determinada porta.
- Quais tipos de filtros de pacotes/firewall estão em uso entre outras funcio- nalida- des.

Além disso, o Nmap vem disponível em duas distribuições, a CUI via console e a GUI via interface gráfica através do Zenmap.

A.4 Subvertendo Sistemas de Detecção de Intrusão

Independente da companhia que desenvolveu, as tecnologias que utiliza ou em qual sistema operacional está rodando, todos os IDSs têm algo em comum: são baseados em regras. Estas regras são o fator que determina se um evento é ou não uma ameaça e qual a sua gravidade. Sabendo disso, os atacantes têm diversos métodos para se proteger durante suas explorações. Este capítulo trata da exploração invisível de portas.

O objetivo das experiências práticas é subverter uma instância do Snort baseando os ataques nas instruções coletadas do livro “Exame de redes com Nmap” [Lyon, 2009] e no referido software de mapeamento de rede, Nmap, a partir de uma máquina dentro da rede interna monitorada pelo IDS.

A.4.1 Experiência prática 1 - Fragmentar pacotes

No que se refere a Sistemas de detecção de intrusão, a fragmentação de pacotes pode ser um problema, particularmente por causa do tratamento de irregularidades entre eles, como a sobreposição das partes e as expirações de montagem de fragmentações são ambíguos e diferem bastante entre as plataformas. Estes fatores fazem com que os IDSs tenham que deduzir como o sistema que recebe os pacotes vai interpretar um pacote.

A.4.1.1 Objetivo

Conseguir informações sensíveis da máquina alvo sem que sejam dispara- dos alertas pelo Snort utilizando a estratégia de fragmentação de pacotes.

A.4.1.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do scan padrão do Nmap através do script: **nmap** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede;

2) **Ataque adaptado:** Com a estratégia de manipulação, foi feito um segundo ataque, desta vez alterado, na tentativa de burlar a(s) regra(s) disparada inicialmente: **nmap -f** <**alvo**>.

A.4.1.3 Conclusão:

A partir das experiências, pôde-se notar que, somente a fragmentação não surtiu efeito na ocultação do ataque, pois ainda surtiram alertas com relação ao mapeamento de portas. E ainda teve um efeito contrário ao esperado: ao invés de minimizar a visibilidade do ataque, acabou causando uma avalanche de alertas causados pela fragmentação dos pacotes, pois o Snort interpretou esses alertas como sendo parte de uma tentativa de ataque de negação de serviço.

A.4.2 Experiência prática 2 - Scans do Protocolo IP

Scans do Protocolo IP permitem que você determine quais protocolos IP (TCP, ICMP, IGMP, etc.) são suportados pelas máquina-alvo.

O scan de protocolo funciona de uma forma similar a um scan UDP. Ao invés de ficar repetindo alternando o campo de número de porta de um pacote UDP, ele envia cabeçalhos de pacote IP e faz a repetição alternando o campo de protocolo IP de 8 bits. Os cabeçalhos normalmente estão vazios, e é nessa característica padrão que o Snort se baseia para detectar este tipo de scan.

A.4.2.1 Objetivo

Demonstrar o comportamento do Snort a partir do scan do protocolo IP feito pelo Nmap. Primeiramente foi feito o scan padrão e em seguida foi executado o mesmo exame manipulado de forma a tentar burlar a regra responsável por disparar alertas em caso de ataque.

A.4.2.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do scan do protocolo IP através do script: **nmap -sO** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede. Seus alertas e resultados serviram como referência posteriormente neste ataque.

2) **Análise:** Após a chamada original a partir do nmap, foi disparado um alerta pelo Snort e a com ele, foi possível obter a regra responsável. Em seguida, tendo a regra, foi possível estudar suas opções e características e articular uma forma de manipular o scan a partir dos mesmos.

3) **Ataque adaptado:** Com a estratégia de manipulação, foi feito um segundo ataque, desta vez manipulado, na tentativa de burlar a regra disparada inicialmente: **nmap -sO** <**alvo**> **–data-length 8**.

A.4.2.3 Conclusão:

Após a modificação do script do scan, incluindo o parâmetro responsável por definir o tamanho do pacote enviado e atribuindo a ele um valor, pode-se perceber que nenhum alerta foi emitido pelo Snort, além dos ruídos da rede, fazendo o ataque passar totalmente invisível aos olhos dos administradores da rede.

No lado do Snort, para prevenir que a regra seja burlada, neste caso poderia ser retirado o parâmetro que determina o tamanho do pacote como 0. Porém, teria que ser avaliada a possibilidade de aumentar falsos positivos devido a isso.

A.4.3 Experiência prática 3 - SCAN nmap XMAS

O ataque “Xmas scan” tem como objetivo diferenciar os status das portas entre “closed” e “open/filtered”. Abaixo podemos observar a chamada original gerada pelo nmap, ainda utilizando pacotes TCP com os sinalizador FPU (FIN, PSH, URG) ligados Apesar de que como todas as portas já estão com status filtered/open, o ataque Xmas scan não se faria necessário neste caso.

A.4.3.1 Objetivo

Demonstrar o comportamento do Snort a partir do scan de natal (Xmas scan) feito pelo Nmap. Primeiramente foi feito o scan padrão e em seguida foi executado o mesmo exame manipulado de forma a tentar burlar a regra responsável por disparar alertas em caso de ataque.

A.4.3.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do Scan de Natal através do script: **nmap -sX** <**alvo**>, onde o alvo foi a máquina de IP 10.0.0.12, o servidor principal da rede. Seus alertas e resultados servirão como referência posteriormente neste ataque.

2) **Análise:** Após a chamada original a partir do nmap, foi disparado um alerta pelo Snort e a partir dele, foi possível obter a regra responsável. Em se- guida, tendo a regra, vai ser possível estudar suas opções e características e articular uma forma de manipular o scan a partir dos mesmos.

3) **Ataque adaptado:** Com a estratégia de manipulação, foi feito um segundo ataque, desta vez manipulado, na tentativa de burlar a regra disparada inicialmente: **nmap -sX** <**alvo**> **–scanflags FINPSH.**

A.4.3.3 Conclusão

Após a modificação do script do scan, incluindo o parâmetro responsável por definir os sinalizadores que devem ser ativados no pacote e atribuindo a ele o valor para desativar os sinalizadores FIN e PSH, pode-se perceber que nenhum alerta foi emitido pelo Snort, a não ser os ruídos da rede, fazendo o ataque passar despercebido.

A.4.4 Experiência prática 4 - Utilizando iscas

O termo isca na área da exploração de vulnerabilidades, diferente da pescaria em que se usa uma isca para atrair a atenção do alvo, remete ao uso de vários pacotes

para distrair o alvo para que o real pacote parte do ataque possa se misturar em meio a uma enxurrada de pacotes “inocentes”.

A.4.4.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente foi feito o scan padrão e em seguida foi executado o mesmo exame manipulado de forma a tentar distrair o Snort e disparar alertas falso-positivos.

A.4.4.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão.

2) **Ataque adaptado:** Com a estratégia de manipulação, foi feito um segundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -D RND:5** <**alvo**>

A.4.4.3 Conclusão

Após a comparação entre os dois resultados, antes e depois da modificação do script, foi possível observar que o comportamento esperado foi atingido, de forma que os alertas gerados pelo ataque real, proveniente do IP original, foram cercados por falsos alertas que apontam para IPs que não são do atacante. Isto, em uma situação real, causaria um alerta para os administradores, porém seria bastante difícil a busca pelo real responsável.

A.4.5 Experiência prática 5 - Simulação de exames de porta

O exame de portas simulando ser feito por um outro host que não o do real atacante. Desta forma, o alerta (que foi gerado) do IDS apontará para um host que foi determinado pelo atacante. Porém, da mesma forma que ele indica um ataque partindo do host simulado, ele retorna as possíveis informações relevantes para o mesmo host que não é o de origem real do atacante, portanto neste tipo de distração não existe a vantagem de conseguir informações sensíveis e úteis sobre o alvo.

A.4.5.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente foi feito o scan padrão e em seguida, foi executado o

mesmo exame manipulado, de forma a tentar distrair o Snort e fazê-lo disparar alertas apontando para uma máquina que não a do real atacante.

A.4.5.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão.

2) **Ataque adaptado:** Com a estratégia de manipulação, foi feito um segundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -Pn -e enp0s8 -S** <**falsa origem**> <**alvo**>, onde o parâmetro -e é a interface a ser usada e a falsa origem é o IP aleatório: 216.20.149.10

A.4.5.3 Conclusão

Após observar ambos os ataques, antes e depois da manipulação através do parâmetro -S, foi possível comprovar que ao fabricar pacotes endereçados por um IP aleatório, o Snort gera alertas falsos já que apontam para uma máquina que não é a do atacante. Neste caso, os alertas são falsos pois, diferente do ataque ocioso, não são aproveitadas informações por parte do atacante, o único objetivo é distrair.

A.4.6 Experiência prática 6 - Exame ocioso

Em 1998 um pesquisador de segurança chamado Antierez desenvolveu uma técnica que permitiria exames de portas de forma não fosse possível para o alvo descobrir diretamente de onde estariam vindo os verdadeiros ataques, ao mesmo tempo que obteria resultados precisos sobre as portas de TCP. Esta técnica ficou conhecida como exame ocioso, pois o atacante forja sua identidade de forma que pareça que uma máquina zumbi inocente fez o exame.

A.4.6.1 Objetivo

Demonstrar o comportamento do Snort antes e depois de um mesmo scan feito pelo Nmap. Primeiramente foi feito o scan padrão e em seguida foi executado o mesmo exame manipulado de forma a tentar distrair o Snort e disparar alertas apontando para uma máquina que não a do real atacante.

A.4.6.2 Execução

1) **Ataque original:** Inicialmente foi executada a chamada original do Scan do Nmap através do script: **nmap -v -Pn** <**alvo**>, onde o alvo é a máquina de IP 10.0.0.12, o servidor principal da rede, -v é para o Nmap exibir mais detalhes e -Pn é para não executar o Ping padrão;

2) **Ataque adaptad**o: Com a estratégia de manipulação, foi feito um se- gundo ataque, desta vez manipulado, na tentativa de distrair o IDS Snort: **nmap -sI** <**máquina zumbi**>**-p 80** <**alvo**>, onde a máquina zumbi é o IP: 10.0.0.18 e o alvo continua sendo 10.0.012;

A.4.6.3 Conclusão

O objetivo do ataque ocioso, é ser a junção entre o uso de iscas e o de simulação de portas, pois é utilizado um IP diferente do atacante, porém ainda assim ele consegue as informações do alvo. Analisando os dados de antes e depois dos ataque, comprova-se que o objetivo é atingido, pois as informações são as mesmas e além disso a identidade do atacante ficou oculta, mesmo que seu ataque tenha gerado alertas.

A.5 Conclusão

A partir dos dados obtidos, e principalmente dos experimentos práticos efetuados, foi possível perceber que, a maior parte das tentativas de obtenção de dados sensíveis, se utiliza de limiares ou brechas presentes na rede devido a omissão durante a configuração. Portanto, executando os mesmos ataques feitos neste estudo, em uma rede real, será possível encontrar as principais brechas e adaptar as regras de segurança do IDS para captar e alertas estes exames, como no caso do exame de natal adaptado, sem os sinalizadores, seria possível adaptar a regra para não exigir que os sinalizadores estejam ligados para disparar alertas.

Porém, também é necessário ter em mente que, mesmo os ataques que são notificados, nem sempre são ataques reais ou apontam para o atacante real, como o Exame ocioso. Para estes casos, o ideal é criar um monitoramento baseado em número de alertas, pois assim, os administradores não vão gastar mais tempo do que o necessário analisando dados, e mesmo assim manteriam a segurança na rede.

Finalmente, ficou evidente que, o IDS deve ser utilizado em conjunto com outras ferramentas, pois apesar de ser efetivo no monitoramento, pode ser otimizado no aspecto de filtragem de falsos alertas e ruídos de rede. Mais especificamente, é imprescindível que ó Snort seja utilizado associado com algum firewall, principalmente para controlar o fluxo de portas que não podem ser fechadas devido ao seu uso.